RETROSPEKTIVE ANALYSE DER AUSBREITUNG UND DYNAMISCHE ERKENNUNG VON WEB-TRACKING DURCH SANDBOXING

TIM WAMBACH

vom Promotionsausschuss des Fachbereichs 4: Informatik der Universität Koblenz–Landau
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation

Universität Koblenz–Landau

Oktober 2018

VORSITZENDES MITGLIED DES PROMOTIONSAUSSCHUSSES:
Prof. Dr. Maria A. Wimmer

VORSITZENDES MITGLIED DER PROMOTIONSKOMMISSION:
Prof. Dr. Dietrich Paulus

BERICHTERSTATTER:
Prof. Dr. Rüdiger Grimm (Universität Koblenz–Landau)
Prof. Dr. Konstantin Knorr (Hochschule Trier)
Prof. Dr. Jan Jürjens (Universität Koblenz–Landau)

DATUM DER EINREICHUNG: 11.04.2018

DATUM DER WISSENSCHAFTLICHEN AUSSPRACHE: 30.10.2018

Tim Wambach: *Retrospektive Analyse der Ausbreitung und dynamische Erkennung von Web-Tracking durch Sandboxing*, Dissertation zur Erlangung des akademischen Grades Doctor rerum naturalium (Dr. rer. nat.).

Für meine Eltern.

KURZFASSUNG

Aktuelle quantitative Analysen von Web-Tracking bieten keinen umfassenden Überblick über dessen Entstehung, Ausbreitung und Entwicklung. Diese Arbeit ermöglicht durch Auswertung archivierter Webseiten eine rückblickende Erfassung der Entstehungsgeschichte des Web-Trackings zwischen den Jahren 2000 und 2015. Zu diesem Zweck wurde ein geeignetes Werkzeug entworfen, implementiert, evaluiert und zur Analyse von 10 000 Webseiten eingesetzt. Während im Jahr 2005 durchschnittlich 1,17 Ressourcen von Drittparteien eingebettet wurden, zeigt sich ein Anstieg auf 6,61 in den darauffolgenden 10 Jahren. Netzwerkdiagramme visualisieren den Trend zu einer monopolisierten Netzstruktur, in der bereits ein einzelnes Unternehmen 80 % der Internetnutzung überwachen kann.

Trotz vielfältiger Versuche, dieser Entwicklung durch technische Maßnahmen entgegenzuwirken, erweisen sich nur wenige Selbst- und Systemschutzmaßnahmen als wirkungsvoll. Diese gehen häufig mit einem Verlust der Funktionsfähigkeit einer Webseite oder mit einer Einschränkung der Nutzbarkeit des Browsers einher. Mit der vorgestellten Studie wird belegt, dass rechtliche Vorschriften ebenfalls keinen hinreichenden Schutz bieten. An Webauftritten von Bildungseinrichtungen werden Mängel bei Erfüllung der datenschutzrechtlichen Pflichten festgestellt. Diese zeigen sich durch fehlende, fehlerhafte oder unvollständige Datenschutzerklärungen, deren Bereitstellung zu den Informationspflichten eines Diensteanbieters gehören.

Die alleinige Berücksichtigung klassischer Tracker ist nicht ausreichend, wie mit einer weiteren Studie nachgewiesen wird. Durch die offene Bereitstellung funktionaler Webseitenbestandteile kann ein Tracking-Unternehmen die Abdeckung von 38 % auf 61 % erhöhen. Diese Situation wird durch Messungen von Webseiten aus dem Gesundheitswesen belegt und aus technischer sowie rechtlicher Perspektive bewertet.

Bestehende systemische Werkzeuge zum Erfassen von Web-Tracking verwenden für ihre Messung die Schnittstellen der Browser. In der vorliegenden Arbeit wird mit *DisTrack* ein Framework zur Web-Tracking-Analyse vorgestellt, welches eine Sandbox-basierte Messmethodik verfolgt. Dies ist eine Vorgehensweise, die in der dynamischen Schadsoftwareanalyse erfolgreich eingesetzt wird und sich auf das Erkennen von Seiteneffekten auf das umliegende System spezialisiert. Durch diese Verhaltensanalyse, die unabhängig von den Schnittstellen des Browsers operiert, wird eine ganzheitliche Untersuchung des Browsers ermöglicht. Auf diese Weise können systemische Schwachstellen im Browser aufgezeigt werden, die für speicherbasierte Web-Tracking-Verfahren nutzbar sind.

ABSTRACT

Current quantitative analyses of web tracking offer no comprehensive overview of its origin, proliferation and development. Through the evaluation of archived websites, this thesis provides a retrospective account of the development history of web tracking between the years 2000 and 2015. For this purpose, a suitable tool was designed, implemented, evaluated and utilised to analyse 10,000 websites. While an average of 1.17 third-party resources were embedded in 2005, the following 10 years saw an increase to 6.61. Network diagrams visualise the trend towards a monopolised network structure in which a single company is able to monitor 80% of internet usage.

Despite numerous attempts to counteract this development using technical measures, only very few self and system-protection systems prove effective, and these are often accompanied by a loss of website functionality or limitations to browser usability. The presented study also establishes that legal regulations offer no adequate protection either. Deficiencies in the fulfilment of data protection obligations are identified on the websites of educational institutions. These are manifested in missing, incorrect or incomplete data protection statements, the provision of which is one of the information obligations of a website provide.

The sole consideration of classic trackers is not sufficient, as demonstrated by another study. By providing functional website components, a tracking company is able to increase coverage from 38% to 61%. This situation is demonstrated using measurements from websites from the health sector and evaluated from a technical and legal perspective.

Existing systemic tools for recording web tracking activities use the interfaces of the browser. This thesis presents *DisTrack*, a framework for web tracking analysis, which follows a sandbox-based measurement methodology. This is a procedure that is used successfully in the dynamic analysis of malware, and which specialises in the detection of side effects on the surrounding system. The behavioural analysis operates independently of the browser programming interface and thus enables a comprehensive examination of browser activity. This allows systemic vulnerabilities in the browser to be identified, which can be utilised by storage-based web tracking.

»It is the struggle itself that is most important. We must strive to be more than we are. It does not matter that we will not reach our ultimate goal. The effort itself yields its own reward.« — Gene Roddenberry

DANKSAGUNG

Mein tiefster Dank gilt meinen Eltern und meinem Bruder für die andauernde Ermutigung, Motivation und Entlastung, was diese Arbeit ermöglicht hat.

Großer Dank gilt außerdem Prof. Dr. Rüdiger Grimm für die engagierte Betreuung der Arbeit, die immer offene Tür bei allen Fragen und die herzliche Aufnahme in die Arbeitsgruppe IT-Risk-Management. Ebenso danke ich Prof. Dr. Konstantin Knorr für die inhaltliche Begleitung der Arbeit und Unterstützung bei Veröffentlichungen.

Sehr herzlich möchte ich mich bei Susanne Krumholz, Tobias Krumholz, Thomas Sader, Christina Huneck und Dr. Laura Schulte für inhaltlichen Diskussionen, Anregungen und Korrekturen bedanken. Für stetige Unterstützung danke ich den derzeitigen und ehemaligen Mitgliedern der Arbeitsgruppe: Dr. Katharina Bräunlich, Dr. Andreas Kasten, Dr. Daniela Simic und Brigitte Jung.

Ein besonderer Dank gilt dem Fachbereich Informatik der Hochschule Trier, insbesondere Prof. Dr. Rainer Oechsle und Prof. Dr. Andreas Künkler, die mich während meines Studium und darüber hinaus gefördert und unterstützt haben. Für die langanhaltende Unterstützung möchte ich mich bei der Gameforge AG und insbesondere bei Alexander Rösner, Tomas Burck und René Fischer bedanken.

Abschließend möchte ich auch der Volkswagenstiftung einen großen Dank aussprechen, deren Förderung diese und viele andere wissenschaftliche Arbeiten ermöglicht hat.

INHALTSVERZEICHNIS

1 EINLEITUNG

Datensammlung

Mit der Nutzung des World Wide Webs geht eine permanente Erfassung, Vermessung und Analyse von erzeugten Nutzungsdaten einher. Für Nutzer ist nicht ersichtlich, mit wie vielen Parteien bei Abruf einer Webseite kommuniziert wird und welche Informationen im Zuge dessen übermittelt werden. So ist in den vergangenen Jahren ein Netz von Trackern entstanden, in dem Internetaktivitäten automatisch und ohne Transparenz mit einer ständig wachsenden Anzahl Dritter geteilt werden.

Datennutzung

Durch Web-Tracking werden Nutzer bei Abruf von Webseiten wiedererkannt und mit bereits gespeicherten Daten in Verbindung gebracht. Die Zuordnung dieser Daten zu Personen ermöglicht den Aufbau von personenbezogenen Datenbanken. Sind diese einmal erhoben, können sie für vielfältige Zwecke eingesetzt werden: Ein typisches Anwendungsbeispiel ist die Generierung personalisierter Werbung.

Schutzlosigkeit

Zu berücksichtigen ist, dass nach der Erhebung kein technischer Schutz vor Zweckentfremdung besteht. Ein schwer überschaubarer Kreis an Unternehmen kann unkontrolliert auf personenbezogenen Daten zugreifen. Ein wesentliches Problem ist, dass eine Überwachung durch Dritte stattfindet, ohne dass sich die Nutzer über diesen Umstand bewusst sind. Eine Anpassung des eigenen Verhaltens ist aufgrund dieser Situation nicht möglich. Infolgedessen ist zu erwarten, dass Nutzer durch ihr Nutzungsverhalten auch Informationen offenbaren, die sie im Bewusstsein dieser Überwachungssituation nicht preisgeben würden.

Auswirkungen

Die erhobenen Daten können über Jahre gespeichert und neuen Algorithmen zugeführt werden. Die langfristigen Konsequenzen sind nicht absehbar, was bei Daten über sensible Lebensbereiche besonders kritisch ist. Neben negativen Auswirkungen auf die Privatheit von Nutzern muss Web-Tracking auch aus der Sicht der Informationssicherheit kritisch beurteilt werden. In der gleichen Weise, wie private Internetnutzer überwacht werden, findet dies auch bei Mitarbeitern statt, die sich zur Erfüllung ihrer Aufgaben im World Wide Web bewegen. Die Art der besuchten Webseiten, getätigte Eingaben und hochgeladenes Material können Informationen über Tätigkeiten und Inhalte offenbaren, deren Weitergabe an Dritte nicht vorgesehen war. Ein solcher unkontrollierter Informationsabfluss kann zu erheblichen und langfristigen negativen Konsequenzen für ein Unternehmen führen.

Bekanntheit

Die Tatsache, dass der Einsatz von Web-Tracking bisher nur wenig mediales Interesse erfahren hat, kann in der harmlosen Außenwirkung begründet sein. Personenbezogene Werbung wird von vielen Nutzern nicht als Problem, sondern als Service verstanden: Generische Werbung wird durch relevante ersetzt. Auf diesem Weg werden Dienstleistungen finanziert, die andernfalls kostenpflichtig wären.

1.1 MOTIVATION

Entwicklung

In welchem Umfang Web-Tracking derzeit eingesetzt wird, zeigt sich durch Messungen und Analysen. Insbesondere seit der Entwicklung von sozialen Netzwerken und der damit einhergehenden Verbreitung von Social-Plugins im Web sind vermehrt Publikationen veröffentlicht worden. Aufgrund der vielfältigen Vorgehensweisen solche Studien durchzuführen, unterscheiden sie sich in Aufbau, Methodik und Zielen. Dies erschwert es, einen Eindruck der Entwicklung über einen größeren Zeitraum zu belegen. Das Aussehen eines solchen Entwicklungsbildes ist unklar und könnte sowohl in der Informatik als auch in anderen Forschungsdisziplinen für informatorische und gesellschaftliche Theorien dienen.

Umgang

Neben der Entstehungsfrage ist auch offen, wie Anbieter von Webseiten mit diesem Thema heute umgehen. Sofern ein Betreiber deutschem Recht unterliegt, müssen Nutzer über den Einsatz von Trackingverfahren aufgeklärt werden. Dies wird über eine Datenschutzerklärung realisiert, die auf der Webseite einsehbar sein und relevante Informationen zum Datenschutz beinhalten muss. Fraglich ist, ob diese stets vorhanden ist und eine vollständige Offenlegung aller eingesetzten Tracker beinhaltet.

Datenschutz

In Datenschutzerklärungen wird nach aktuellem Stand nur über klassische Tracker aufgeklärt. Mit der gleichen Funktionalität, die Tracker in Webseiten integrieren kann, lassen sich auch andere Bestandteile einbinden. Beispiele sind Schriftarten, Videos und Skript-Bibliotheken. Auch wenn diese Einbettungen auf den ersten Blick einen funktionalen Zweck erfüllen, ist zu hinterfragen, wie diese technisch und rechtlich zu bewerten sind. Je nach Art und Umfang der Datenweitergabe müssen sie wie klassische Tracker berücksichtigt und innerhalb der Datenschutzerklärung genannt werden.

Messwerkzeuge

Die beschriebenen Messungen können durch verschiedene Messwerkzeuge realisiert werden. In der Regel wird ein Browser eingesetzt, der zu diesem Zweck modifiziert oder erweitert wird. Bislang ist kein strukturierter Versuch unternommen worden, diese Messungen außerhalb des Browsers durchzuführen. Diese Arbeit geht der Frage nach, ob eine Beobachtung des Browserverhaltens Rückschlüsse über den Einsatz von Web-Tracking gibt. Ein solches System könnte zukünftig sowohl zur Analyse von Webseiten als auch zur Überprüfung des Browsers eingesetzt werden.

1.2 ZIELSETZUNG

Der vorherige Abschnitt zeigt bestehende offene Fragen auf. Im Folgenden sind die daraus resultierenden Forschungsfragen umrissen, die in dieser Dissertation behandelt werden.

FORSCHUNGSFRAGE RQ-1

Vergangenheit

In welcher Weise hat sich Web-Tracking in den vergangenen Jahren ausgebreitet?

Allgemein wird von einer Zunahme von Web-Tracking im Web ausge-

gangen. Bestehende wissenschaftliche Studien analysieren nur einen kurzen Zeitraum oder behandeln speziellere Fragestellungen. In den Kapiteln 3-6 wird sich mit der Frage befasst, wie der Entwicklungsverlauf rückwirkend erfasst werden kann.

FORSCHUNGSFRAGE RQ-2

Gegenwart I

In welchem Umfang findet Web-Tracking auf Webseiten des tertiären Bildungsbereiches statt und wie wird dieses in den Datenschutzerklärungen berücksichtigt?

Verwandte Arbeiten stützen die Vermutung, dass viele Webseitenbetreiber nicht mit den rechtlichen Anforderungen vertraut sind, die mit der Bereitstellung ihrer Dienste einhergehen. In Abschnitt 8.1 wird eine Prüfung von Institutionen beschrieben und durchgeführt, die am ehesten eine Konformität erwarten lassen.

FORSCHUNGSFRAGE RQ-3

Gegenwart II

In welchem Ausmaß finden Einbettungen von Drittinhalten auf Webseiten im Gesundheitswesen statt und was sind die Konsequenzen?

Auf Webseiten mit Gesundheitsthemen wird ein höheres Maß an Privatheit erwartet. Dies betrifft auch die Webauftritte von Krankenhäusern und Klinken. Fraglich ist, ob sich diese Erwartung mit der Realität deckt. In Abschnitt 8.2 wird eine rechtliche und technische Analyse vorgenommen, um das Trackingpotenzial von Einbettungen zu bewerten, die aus funktionalen Gründen in die Webseite integriert sind.

FORSCHUNGSFRAGE RQ-4

Zukunft

Welchen Mehrwert bietet die verhaltensbasierte Analyse des Browsers zur Erkennung von Web-Tracking-Methoden? ?

Werkzeuge, welche eine Verhaltenserkennung von Anwendungen durchführen, sind im Bereich der Schadsoftwareanalyse verbreitet. In Hinblick auf speicherbasierte Trackingverfahren zeigen sich Parallelen, welche die Frage aufwerfen, ob diese Methodik auch bei Browsern eingesetzt werden kann. In Kapitel 10 wird ein solches Werkzeug implementiert und in Kapitel 11 evaluiert.

Abgrenzung

Zusammenfassend wird die Entwicklung des Web-Trackings in der Vergangenheit sowie in der Gegenwart analysiert. Schließlich wird eine neue Analysemethodik für die Zukunft präsentiert.

Wie diese Daten durch die Tracker und Werbeanbieter verarbeitet bzw. wie aus diesen Daten Informationen gewonnen werden, wird nicht genauer behandelt, weil dies Themen der Forschungsgebiete Data-Warehouse und Data-Mining (auch: „Big Data“) sind. Für Informationen zu diesen Themen sei beispielsweise auf Wills und Tatar [200] verwiesen.

Die Methodiken werden passend zur jeweiligen Forschungsfrage erarbeitet. Jede Problemstellung wird in einem separaten Abschnitt methodisch betrachtet.

FORSCHUNGSFRAGE RQ-1: Abschnitt 4.2.

FORSCHUNGSFRAGE RQ-2: Abschnitt 8.1.3.

FORSCHUNGSFRAGE RQ-3: Abschnitt 8.2.3.

FORSCHUNGSFRAGE RQ-4: Abschnitt 10.2.

Design Science Research

Die Forschungsfragen RQ-1 und RQ-4 werden auf Basis von *Design Science Research* behandelt. „Design Science“ entstand aus den Ingenieurswissenschaften und wird in der Wirtschaftsinformatik als Gegenstück zur verhaltenswissenschaftlichen (behavioral-science) Methodik verstanden, welche ihren Ursprung wiederum in den Naturwissenschaften hat [74, S. 76]. Vaishnavi und Kuechler stellten 2004 eine Methodik [187] vor, welche die Unterscheidung zwischen Naturwissenschaft und „Design Science“ von Simon aus dem Jahr 1996 konkretisiert [166]. Vishnavi und Kuechler verwenden dabei ein Modell (vgl. Abbildung 1.1), welches die folgenden fünf Stadien umfasst:

1. Erkennen des Problems (Awareness of Problem): Erkennen und analysieren des Problems und der Ausgangslage. Ein wesentliches Ergebnis dieses Schrittes ist eine detaillierte Problembeschreibung.
2. Lösungsvorschlag (Suggestion): Die Anfertigung eines Lösungskonzeptes und die Generierung von Prüfkriterien soll in diesem Schritt durchgeführt werden.
3. Entwicklung (Development): Es folgt die Umsetzung des aus der vorherigen Phase entstandenen Lösungskonzeptes. Der Output dieses Schrittes wird als Artefakt bezeichnet.
4. Bewertung (Evaluation): Anschließend findet eine Prüfung des entwickelten Artefaktes anhand vorher festgelegter Kriterien statt.
5. Zusammenfassung (Conclusion): Eine abschließende Analyse der Resultate bildet das Gesamtergebnis der Arbeit.

Iterationen

Es muss beachtet werden, dass sämtliche Schritte stets zu einem besseren Verständnis des Problems beitragen können. Daher ist dieses Vorgehen als Zyklus anzusehen, bei dem nachfolgende Phasen die Ausgangsbasis ändern und so weitere Iterationen bewirken können.

Hevner et al.

Die Vorgehensweise mit *Design Science in Information Systems Research* von Hevner et al. aus dem Jahr 2004 [74] ist in ähnlicher Weise strukturiert. Im Vordergrund steht jedoch die Generierung eines (IT-)Artefakts. Dabei werden die Menschen (People), die Organisation (Organization) und die technologische Ausrichtung (Technology) als „Business Needs“ in den Entwicklungs- und Evaluierungszyklus eingebracht [74, S. 80].

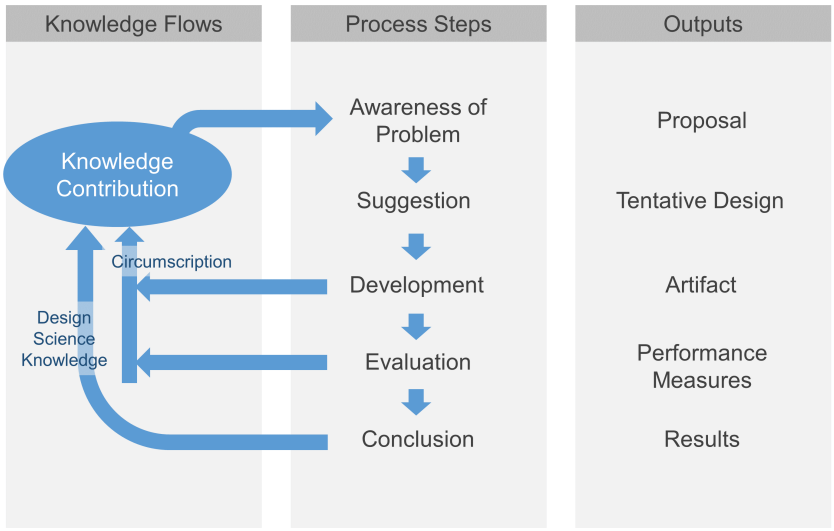

Abbildung 1.1: Das Design Science Research (DSR) Prozessmodell nach Vaishnavi und Kuechler [187].

1.4 Aufbau der Arbeit

Lesefolge

Die vorliegende Arbeit besteht aus drei Teilen, die sich der Vergangenheit, Gegenwart und Zukunft widmen. Aufgrund dieser chronologischen Anordnung wird grundsätzlich eine sequenzielle Lesefolge empfohlen. Alternative Lesefolgen können in Abbildung 1.2 betrachtet werden.

Zeit und Ansprache

Sofern nicht anders angegeben ist, beziehen sich zeitliche Verweise auf das Jahr der Einreichung dieser Arbeit (2018). Aus Gründen der besseren Lesbarkeit gelten im Folgenden sämtliche Personenbezeichnungen gleichwohl für jede Form von Geschlecht.

1.5 Veröffentlichungen

Veröffentlichungen und Zitate

Dieser Dissertation sind Veröffentlichungen vorausgegangen, die als Quellen genutzt werden. Textabschnitte werden nur dann teilweise oder vollständig übernommen, wenn sie ausschließlich aus der Hand des Autors stammen. Abschnitte von Co-Autoren werden hingegen zitiert und entsprechend ausgezeichnet. Im Folgenden sind diese Veröffentlichungen chronologisch aufgelistet:

- Wambach, Tim (2017): Ökonomisierung von Nutzerverhalten – historische Entwicklung und aktueller Stand [190],
- Wambach, Tim / Schulte, Laura / Knorr, Konstantin (2016): Einbettung von Drittinhalten im Web [194],

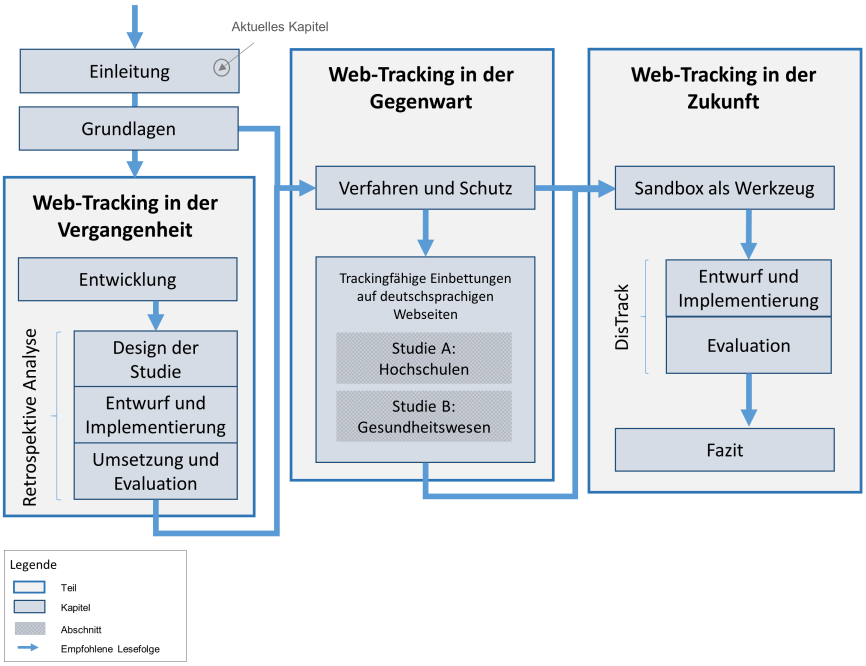

Abbildung 1.2: Aufbau der Arbeit.

- Wambach, Tim / Bräunlich, Katharina (2016): The Evolution of Web-Tracking [192],
- Wambach, Tim / Bräunlich, Katharina (2016): Retrospective Study of Third-Party Web Tracking [191],
- Wambach, Tim / Knorr, Konstantin (2015): Technische Prüfung der Datenschutzerklärungen auf deutschen Hochschulwebseiten [193],
- Wambach, Tim (2015): Dynamische Trackererkennung im Web durch Sandbox-Verfahren [189].

2

GRUNDLAGEN

Zusammenfassung: In diesem Kapitel werden Grundlagen und weiteres Hintergrundwissen vermittelt. Nach einem kurzen Einstieg in das Thema wird Tracking aus verschiedenen Perspektiven betrachtet. Dies umfasst die Abgrenzung zu Werbung, die Betrachtung von Tracking als Privatheitsproblem und den Blick aus der Sicht der Informationssicherheit. Abschließend werden wichtige Begriffe definiert und erklärt.

2.1 TRACKING-VERFAHREN

Begriff

Als Web-Tracking wird die Verfolgung der Aktivitäten von Nutzern im Web bezeichnet. Für Mayer und Mitchell [115] steht der Begriff sinnbildlich für das Sammeln von Verhaltensdaten. Zum Aufbau einer Sammlung müssen die Einzelaktivitäten gruppiert und einem Nutzer zugewiesen werden.

Umsetzung

Die Gruppierung kann auf unterschiedliche Weise technisch umgesetzt werden. Bei zustandsbasierten Tracking-Verfahren findet eine Markierung des Nutzersystems statt. Eine solche Markierung kann durch ein HTTP-Cookie erreicht werden. Darüber hinaus existieren subtilere Methoden, die nur schwierig vom Nutzer erkannt werden können. Neben weiteren offiziellen Speicherformen des Browsers lassen sich sowohl Eigenarten im HTTP-Protokoll als auch Caching-Verfahren zur Markierung des Systems nutzen. Bei zustandslosen Verfahren erfolgt die Wiedererkennung des Nutzers durch das Verhalten des verwendeten User-Agents. Besonderheiten des genutzten Browsers und des darunterliegenden Systems ermöglichen die Unterscheidung der Aktivitäten von anderen Nutzern. Beispielsweise führt die Installation selten genutzter Schriftarten dazu, Systeme von anderen unterscheidbar zu machen, was bei Fingerprint-basierten Tracking-Verfahren ausgenutzt wird.

Datenschutz

Die Zuordnung von gruppierten Daten zu einem Nutzer bewirkt einen Personenbezug. Daraus ergeben sich möglicherweise datenschutzrechtliche Konsequenzen. So kann das Nutzungsverhalten sensible Informationen offenbaren wie zum Beispiel persönliche Vorlieben und Einkaufsentscheidungen beim Surfen und Online-Shopping, deren Weitergabe an Dritte nicht im Interesse des Nutzers steht. Dies muss kritisch bewertet werden, sofern der Nutzer über die Weitergabe nicht vollständig informiert ist und ihr zustimmt.

Gründe

Der Einsatz von Web-Tracking kann verschiedene Gründe haben. Die Nutzung der Informationen zur Generierung von personalisierter Werbung stellt einen der Hauptgründe dar. Die Verwendung dieser gesammelten Hin-

tergrundinformationen ermöglicht die Einblendung von Produktwerbung, die besonders auf die Interessen des jeweiligen Nutzers zugeschnitten ist.

2.2 SCHUTZMASSNAHMEN

Anforderungen

Aus der Perspektive der Trackinganbieter muss das genutzte Trackingverfahren unter anderem die Anforderung der Stabilität erfüllen. Das bedeutet, dass das verwendete Verfahren über einen längeren Zeitraum zuverlässig arbeitet. Dieser Anforderung stehen Schutzmaßnahmen im Weg, die vom Nutzer selbst genutzt werden oder bereits systemisch in den Browser integriert sind.

Schutzkonzepte

Ein häufig eingesetztes Schutzkonzept stellt die Blockierung dar. Dabei wird beispielsweise das Setzen von Cookies durch Trackinganbieter im Browser unterbunden. Auch die gezielte Blockierung von Netzwerkverkehr, die Deaktivierung von Skriptausführungen sowie die Unterdrückung von Popups kommen dabei in Betracht. Um diesen Schutz zu automatisieren, werden Erweiterungen des Browsers eingesetzt, die üblicherweise von Drittanbietern stammen. Ein solcher präventiver Schutz kann durch reaktive Maßnahmen wie das regelmäßige Löschen von Cookies, des Caches und Browserverlaufs ergänzt werden. Diesem Selbstschutz stehen Systemschutzkonzepte gegenüber: Durch Maßnahmen wie P3P und Do-Not-Track soll mehr Transparenz in der Erhebung und Weitergabe von personenbezogenen oder -beziehbaren Daten erzielt und die Nutzerwünsche dem Server übermittelt werden.

Selbst- und Systemschutz

Sowohl bei Selbst- als auch bei Systemschutzlösungen zeigen sich Probleme. So ist entweder die erzielte Schutzwirkung als gering anzusehen, oder die aus dem Schutz entstehenden Nachteile stellen eine starke Beeinträchtigung der Nutzbarkeit dar. Dies stellt vor allem unerfahrene Nutzer vor Probleme, was sich schwächend auf die Akzeptanz dieser Lösungen auswirkt. Lösungen in Form von Browsererweiterungen durch Drittanbieter müssen stets kritisch bezüglich ökonomischer Interessen hinterfragt werden. Bei den Systemschutzlösungen gibt es bislang keine rechtlichen Vorgaben, was zu einer geringen Bereitschaft zu deren Umsetzung durch die Anbieter führt. Die bloße Berücksichtigung eines freiwilligen Standards ist nur ein geringer Anreiz, um auf die Entlohnung durch Tracking- und Werbefirmen zu verzichten.

Ausblick

Weitere Informationen zu eingesetzten Technologien im Web-Tracking und Schutzmaßnahmen sind in Kapitel 7 zu finden.

2.3 WEB-TRACKING UND WERBUNG

Werbemarkt

Seit der Jahrtausendwende hat sich das Internet zu einem wesentlichen Bestandteil unseres Alltags entwickelt: Während im Jahr 1997 das Internet nur von 2 % der Weltbevölkerung genutzt wurde, ist dieser Anteil bis zum Jahr 2016 auf 46 % angestiegen [86]. Dieser hohen Nachfrage steht eine Viel-

falt an Angeboten gegenüber: Soziale Netzwerke, Videoplattformen, Online-Marktplätze, private Blogs, etc. Diese Angebote können durch Einbindung von Werbung finanziert werden. Es ist daher wenig überraschend, dass die Umsätze in diesem Markt stetig angestiegen sind:

- Die Gesamtumsätze der Onlinewerbung in Deutschland lagen im Jahr 2008 bei mehr als 3,1 Milliarden Euro und 2015 wurden Umsätze im Wert von 6,55 Milliarden Euro prognostiziert [146].
- Der Werbeumsatz von Google lag im Jahr 2013 bei ca. 51,07 Milliarden US-Dollar (ca. 48,5 Milliarden Euro, Kurs 2016) [5].
- Im Jahr 2014 wurde ein weltweiter Umsatz von 132 Milliarden US-Dollar (ca. 124 Milliarden Euro, Kurs 2016) für das Jahr 2015 prognostiziert und ist damit im Vergleich zum Jahr 2011 um zwei Drittel gewachsen [182].

Vergleich zu anderen Medien

Im Vergleich dazu: Die Bruttowerbeaufwendungen der Printmedien in Deutschland beliefen sich im Jahr 2015 auf ca. 4,67 Milliarden Euro bei Zeitungen, 3,48 Mrd. bei Publikumszeitschriften und ca. 0.4 Mrd. bei Fachzeitschriften [131]. Hierbei ist ein Trend zur personalisierten Werbung zu erkennen. Dieser folgt der Erwartung, dass Werbung, die gezielt auf die Interessen des Webseitenbesuchers abgestimmt ist, diesen stärker beeinflussen kann.

Datenquellen

Die Selektion personenbezogener Werbung erfordert eine Datenbasis von Verbrauchs- und Nutzungsdaten. Die vom Webseitenbesucher aufgerufenen Webseiten stellen eine solche Datenbasis dar. Die Qualität der Daten kann unter Verwendung weiterer Datenquellen verbessert werden: In Betracht kommt hier unter anderem die Social-Media-Profile. Am Beispiel Facebook ist zu sehen, wie exakt Werbetreibende die Zielgruppen bestimmen können, wodurch sich die Art und Intensität der gespeicherten Daten offenbart. Beispiele für Daten, die Facebook für eine Zielgruppenbildung nutzt bzw. Werbetreibenden zur Verfügung stellt, sind[1]:

- Nutzer, die sich wahrscheinlich politisch betätigen,
- Nutzer, die für wohltätige Zwecke gespendet haben,
- Nutzer, die Guthaben auf der Kreditkarte haben.

Auch Google bietet eine sehr feingranulare[2] Selektion der Zielgruppen an.

2.4 Web-Tracking als Privatheitsproblem

Definition

Malandrino et al. [112] verweisen bei der Definition von Privacy auf Westin [198]: „right to prevent the disclosure of personal information“. Diese basiert auf einer Definition von Warren und Brandeis [195] aus dem Jahr 1890: „right to be let alone“.

1 Tischbein, Katharina 2016: 98 Daten, die Facebook über dich weiß und nutzt, um Werbung auf dich zuzuschneiden. https://netzpolitik.org/2016/98-daten-die-facebook-ueber-dich-weiss-und-nutzt-um-werbung-auf-dich-zuzuschneiden/, abgerufen am 17.01.2017.

2 https://developers.google.com/adwords/api/docs/appendix/codes-formats, abgerufen am 17.01.2017

Profilbildung

Web-Tracking wird häufig als Problem der Privatheit (Privacy) und des Datenschutzes verstanden. Ein grundsätzliches Ziel von Web-Tracking ist die Ansammlung und Verknüpfung von Daten, die für eine Profilbildung genutzt werden. Malandrino et al. stellen fest, dass gewisse Formen von Datennutzung als schädlich betrachtet werden müssen:

> „Overall, collection, processing and dissemination of personal information can raise serious privacy issues among users when they go online, for a variety of daily activities [...] "
> Quelle: [112, S. 1].

Mayer und Mitchell [115] abstrahieren konkrete Gefahren und beschreiben die vielfältigen Möglichkeiten, einen Schaden für den betroffenen Nutzer herbeizuführen – auch ohne diese konkret zu benennen. Arten von Schäden, die aus einem Missbrauch der Daten hervorgehen, können nach den Autoren physisch, psychologisch und ökonomisch sein.

Bedrohungen

Schneider et al. [161, S. 58ff] führen im Web-Tracking-Report von 2014 sehr ausführlich Bedrohungsaspekte durch Erhebung und Verarbeitung von personenbeziehbaren Daten aus. Genannt werden: Stalking, Verlust der Freiheit, Schwächung der Demokratie, Manipulation von Bürgern und Filterung von Informationen. Es folgen einige Beispiele:

KREDITWÜRDIGKEIT. Online-Aktivitäten können einen direkten Einfluss[3] auf die Kreditwürdigkeit einer Person ausüben. Bujlow et al. [28] erläutern, wie das Unternehmen *Kreditech* Daten von Facebook, eBay und Amazon.com verwendet, um Ausfallwahrscheinlichkeiten zu ermitteln. Sie beschreiben einen konkreten Fall, in dem das Kreditkartenlimit des Nutzers durch Nutzung eines Onlinemarktes gesenkt wurde. Bujlow et al. führen außerdem aus, wie diese Daten in gleicher Weise bei Abschluss von Versicherungsverträgen, also zur Berechnung von Eintrittswahrscheinlichkeiten, genutzt werden können.

PREISDISKRIMINIERUNG. In der Weise, wie Werbung profilbasiert selektiert werden kann, lassen sich auch Preise individuell auf Basis von Besucherinformationen anpassen. Die Nutzung des Standortes eines Besuchers zur Ermittlung eines individuellen Preises wird von Mikians et al. [119] näher beschrieben. Bisher wurden nur wenige wissenschaftliche Studien durchgeführt; einzelne Erfahrungsberichte[4] zeigen bereits heute Anpassungen der Preise für Urlaubsreisen auf Basis von Browserinformationen.

ÜBERWACHUNG. Im Jahr 2015 präsentierten Englehardt et al. eine Studie [49] zur Massenüberwachung durch Geheimdienste auf Basis von Web-Tracking: „For non-U.S. individuals, our data indicates that the agency could have access to a majority of a person's browsing history

3 http://money.cnn.com/2013/08/26/technology/social/facebook-credit-score/index.html?hpt=hp_t2, zuletzt abgerufen am 30.11.2017

4 http://www.elliott.org/blog/are-online-travel-agencies-quoting-higher-rates-because-of-your-web-cookies/, abgerufen am 30.11.2017

(71.3% of visits in Ireland and 61.1% of visits in Japan), without ever collecting data outside the United States.“, Quelle: [49, S. 9]. Die möglichen Konsequenzen aus geheimdienstlichen Überwachungsmöglichkeiten sind seit den Enthüllungen von Edward Snowden aus dem Jahr 2013 ein Teil des öffentlichen Diskurses.

Abstrakte Gefahren

Mayer und Mitchell [115] stellen einige Fragen zur Diskussion, die aktuell als unbeantwortet oder nur teilweise als beantwortet gelten: zum Beispiel, welcher Teil des Werbemarktes wirklich auf Web-Tracking angewiesen ist und wie die Daten verwendet werden. Des Weiteren wird hinterfragt, ob ein gewisser Teil der Nutzer nicht eher auf Inhalte verzichten würde, wenn sie sich des Privatheitsproblems stärker bewusst wären.

2.5 Web-Tracking in der Informationssicherheit

Gefahren

Die unkontrollierte Weitergabe von Informationen ist als Vertraulichkeitsproblem grundsätzlich auch der Informationssicherheit zuzuordnen. Durch Web-Tracking findet, wissentlich oder unwissentlich, eine Weitergabe von Nutzungsinformationen an Dritte statt, die innerhalb von Risikoanalysen Berücksichtigung finden müssen. Eine Betrachtung von nationalen und internationalen Lehrbüchern zur IT- und Informationssicherheit wie von Eckert [45] und Stewart et al. [174] zeigt, dass Web-Tracking bisher kaum als Sicherheitsproblem wahrgenommen wird. Bujilow et al. stellen in einer Untersuchung [28] von 2015 fest, dass der Hauptteil der Forschung zu diesem Thema in den Jahren 2012 bis 2014 entstanden ist.

Grundschutz

Der IT-Grundschutz-Katalog [29], der vom Bundesamt für Sicherheit in der Informationstechnik als Leitfaden zur Identifikation und Umsetzung von Sicherheitsmaßnahmen in Unternehmen ausgegeben wurde, listet Web-Tracking im Maßnahmenkatalog unter M2.488[5]. So soll die Weitergabe der Besucherdaten an Dritte aus technischen wie aus rechtlichen Gründen beachtet werden. Dementsprechend steht Web-Tracking und damit verbundene technischen Maßnahmen als ein Thema der Informationssicherheit außer Frage. In der Neufassung[6] von 2018 findet keine Erwähnung mehr von Web-Tracking statt. Dabei zeigen sich viele Gründe, diesem Thema eine deutlich größere Aufmerksamkeit zu schenken und stärker in der Informationssicherheit zu thematisieren. Einige Gründe werden im Folgenden beispielhaft ausgeführt:

Informationsabfluss von Unternehmensinterna. Die Nutzung von persönlichen Interessen und Vorlieben mittels Web-Tracking ist bekannt und offensichtlich. Dabei wird wenig berücksichtigt, dass diese Informationen auch im geschäftlichen Umfeld gesammelt werden, während Mitarbeiter als Nutzer im Web aktiv sind. Recherchen auf

5 https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m02/m02488.html, zuletzt abgerufen am 28.11.2017.

6 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2018.pdf, abgerufen am 25.02.2018

speziellen Webseiten oder Suchen nach Produkten offenbaren dabei strategische und somit vertrauliche Informationen. Insbesondere im Fall einer Monopolisierung des Web-Trackings, bei der nur wenige Unternehmen Zugriff auf große Mengen von Nutzungsinformationen haben, ermöglicht eine feingranulare Überwachung einzelner Mitarbeiter eines Unternehmens. Während der Browserverlauf eines Mitarbeiters ohne Zweifel als schutzwürdig eingestuft wird, wird vernachlässigt, in welchem Ausmaß Drittparteien bereits über diese Informationen verfügen und frei verarbeiten können. Englehardt et al. [49] zeigen, wie die Betreiber von Werbenetzwerken in der Lage sind, 62-73 % des Browserverlaufs zu rekonstruieren. Aus diesem Grund muss der Schutz vor Profilbildung eine zentrale Stellung in jeder Unternehmensrichtlinie zur Informationssicherheit einnehmen.
In einem Videobeitrag[7] des 33. Kongress des Chaos Computer Clubs in Hamburg wurde demonstriert, wie aus vermeintlich anonymen Daten, die von Werbeunternehmen bezogen wurden, vertrauliche Informationen aus der Korrespondenz einer Polizeidienststelle entnommen werden konnten.

SCHADSOFTWARE DURCH WERBUNG. Nutzungsbasierte Werbung, in Publikationen auch häufig als Online-Behaviorbased-Advertising (OBA) bezeichnet, wird üblicherweise dynamisch erzeugt. Dies bedeutet, dass die eingeblendete Art der Werbung erst durch das Profil des Webseitenbesuchers selektiert wird. Aus einer größeren Anzahl verschiedener Unternehmen wird die passende Werbung für den jeweiligen Benutzer ausgewählt. Die Hürden, eigene Werbeinhalte zu schalten sind, abgesehen von finanziellen Aufwendungen, gering.

Diese Werbenetzwerke werden auch zur Platzierung von Schadsoftware auf Webseiten genutzt[8]. Aufgrund der zunehmenden Häufigkeit hat sich der Begriff „Malvertising“ etabliert – zusammengesetzt aus Advertising (Werbung) und Malware (Schadsoftware). Detaillierte Analysen finden sich unter anderem bei Sood und Enbody [169], Provos et al. [148] sowie bei Xing et al. [204].

IDENTITÄTSDIEBSTAHL. Die Einbettung von Skripten ermöglicht Web-Trackern die Ausführung beliebiger JavaScript-Programme im Browser der Benutzer. Dies gibt ihnen weitreichende Freiheiten, wie beispielsweise die Überwachung von Aktivitäten der Maus und Tastatur des Besuchers während seines Webseitenbesuchs. Eine solche nachträglich durchgeführte Analyse des Verhaltens heißt Session-Replay. Es ist zu berücksichtigen, dass diese Aufzeichnung in der Regel alle Aktivitäten einschließt – auch die Eingabe von persönlichen Daten oder von Passwörtern. Im November 2017 publizierten[9] Englehardt et al.

7 https://media.ccc.de/v/33c3-8034-build_your_own_nsa, abgerufen am 30.11.2017.
8 https://www.bleepingcomputer.com/news/security/crooks-created-28-fake-ad-agencies-to-disguise-massive-malvertising-campaign/, abgerufen am 27.01.2018.
9 https://freedom-to-tinker.com/2017/11/15/no-boundaries-exfiltration-of-personal-data-by-session-replay-scripts/, zuletzt abgerufen am 28.11.2017.

eine Analyse der Weitergabe von Passwörtern an Werbebetreiber während der Nutzung einer Webseite durch den Besucher. Durch Skripte, die während des Besuches im Hintergrund aktiv sind, wurden alle Nutzereingaben unmittelbar an die Drittpartei übermittelt. Eine missbräuchliche Nutzung dieser Daten wurde bislang noch nicht nachgewiesen.

2.6 BEGRIFFSDEFINITIONEN

Im Folgenden werden Begriffe definiert, die innerhalb der Arbeit häufig eingesetzt werden, aufgrund einer Mehrdeutigkeit klargestellt werden müssen oder auf sonstige Weise unklar sein könnten.

User-Agent

Browser

Der Begriff Browser oder Webbrowser steht in dieser Arbeit stellvertretend für alle Arten von Clients, die eine Webanfrage starten und Antworten der Gegenstelle verarbeiten können. Die Verwendung des Begriffes orientiert sich an der Definition von „user agent“ (UA) in RFC 2616 [121]: „The client which initiates a request. These are often browsers, editors, spiders (web-traversing robots), or other end user tools“.

Aufbau

Grosskurth und Godfrey [66] demonstrieren eine Referenzarchitektur für einen Standard-Browser (bzw. User Agent im Sinne der RFC 2616) und zeigen, wie dieses Modell auf die zu dieser Zeit marktüblichen Browsern angewendet werden kann. Dabei findet eine Unterteilung in die folgenden drei Hauptkomponenten statt:

BENUTZEROBERFLÄCHE. Über das User-Interface wird die Kommunikation mit dem Nutzer ermöglicht und zeigt die Ergebnisse der Verarbeitung an.

RENDERING-MODUL. Dieses dient der Umwandlung (Parsing) der geforderten Ressourcen (z. B. HTML-Quelltext) in eine Baumstruktur (DOM-Tree) und der abschließenden Aufbereitung zur visuellen Darstellung (Painting).

BROWSER-MODUL. Das Browser-Modul bietet abstrakte Methoden für die Benutzerschnittstelle sowie für das Rendering-Modul. Es dient damit als Zwischenglied.

World Wide Web

WWW

Eine Arbeit zum Thema Web-Tracking befasst sich stets mit Web-Technologien. An der Weiterentwicklung sind maßgeblich das World Wide Web Consortium (W3C)[10] und die Internet Engineering Task Force (IETF)[11] beteiligt.

10 https://www.w3.org/, zuletzt abgerufen am 21.10.2017.
11 https://www.ietf.org/, zuletzt abgerufen am 21.10.2017.

Die in dieser Arbeit verwendete Terminologie beruht auf Definitionen aus veröffentlichten Standards vom W3C und den RFCs[12] (Request for Comments) der IETF.

JavaScript

Entstehung

Seit der Einführung[13] im Browser *Netscape Navigator* im Jahr 1995 unter dem Namen „LiveScript“ wurde die heute als JavaScript bekannte Programmiersprache stetig weiterentwickelt. Da jeder Browserhersteller eigene Interpretationen bei der Semantik von Befehlen vornahm, sollte mittels des standardisierten ECMAScript eine Harmonisierung erreicht werden. Dieser Versuch zeigte sich erfolgreich: Die Version 6 wird in allen gängigen Browsern unterstützt[14]. JavaScript ist seit 2017 in Version 8 verfügbar[15].

Möglichkeiten

Mittels APIs (Schnittstellendefinitionen) und vordefinierter Objekte wird die Interaktion zwischen JavaScript-Code und Browser ermöglicht. Das `document`-Objekt[16] erlaubt die Erstellung, Modifikation und Löschung von Inhalten auf einer Webseite. Infolgedessen kann mittels `document.write` ein Text in die Webseite eingefügt werden. `Document.cookies` erlaubt den Zugriff auf die gespeicherten Cookiedaten. Frameworks wie AngularJS oder jQuery erleichtern den Entwickleerrn von JavaScript-Anwendungen den Umgang. So bieten diese Bibliotheken Lösungen für gängige Probleme an, die ansonsten vom Entwickler selbst implementiert werden müssten.

Sicherheit

Grundsätzlich sind direkte Zugriffe auf das darunterliegende System, beispielsweise auf Dateien, nicht möglich und werden durch eingebaute Sicherungen in Browsern verhindert. Die Ausnutzung von Schwachstellen hingegen kann einen Ausbruch aus diesem Sicherungssystem[17] zur Folge haben.

Weitere Definitionen

URL/URL/URN. Die Unterscheidung zwischen URI, URL und URN ist in RFC 3986 [20] definiert. Der Überbegriff URI (Uniform Resource Identifier) unterteilt sich in die Gruppen URL (Uniform Resource Locator) [18] und URN (Uniform Resource Name) [157]. Während die URI die Zugriffsart und den Speicherort (im Netzwerk) beschreibt, dient die URN der eindeutigen Namensgebung. Eine Erweiterung im Jahr 2005 von der URI zur IRI (Internationalized Resource Identifiers) erlaubt die Nutzung von Zeichen außerhalb des ASCII-Satzes gemäß Spezifikation RFC 3987 [43].

12 https://www.ietf.org/standards/rfcs/, abgerufen am 19.03.2018.
13 http://archive.oreilly.com/pub/a/javascript/2001/04/06/js_history.html, abgerufen am 15.03.2018.
14 http://kangax.github.io/compat-table/es6/, abgerufen am 15.03.2018.
15 https://www.ecma-international.org/ecma-262/8.0/index.html, abgerufen am 15.03.2018.
16 https://www.w3schools.com/jsref/dom_obj_document.asp, abgerufen am 15.03.2018.
17 https://www.mozilla.org/en-US/security/advisories/mfsa2009-41/, abgerufen am 15.03.2018.

RESSOURCE. Eine Web-Ressource gemäß dem HTTP 1.1 und definiert in RFC 2616 [121]: „A network data object or service that can be identified by a URI [...]“. Bezogen auf die vorliegende Arbeit handelt es sich um alle Elemente, die durch einen Ladevorgang, adressiert durch eine URI, in einer Webseite eingebunden sind.

DRITTPARTEI. An der Kommunikation über ein Medium sind Sender und Empfänger beteiligt. Im Falle von Web-Tracking tritt der Tracker als ein dritter Kommunikationspartner in Erscheinung. Der Begriff Drittpartei umfasst diesen und alle weiteren Teilnehmer, die zu einer Überwachung der Kommunikation in der Lage sind.

(HYPER)LINK. Eine Definition von Links findet sich in RFC 5988 [135]: „In this specification, a link is a typed connection between two resources that are identified by Internationalised Resource Identifiers (IRIs) [RFC3987], and is comprised of: A context IRI, a link relation type (Section 4), a target IRI, and optionally, target attributes“.

Teil I

WEB-TRACKING IN DER VERGANGENHEIT

Nachdem die Frage nach einem Gesamtbild zu Web-Tracking als Forschungslücke identifiziert wurde, wird Methodik und Design einer retrospektiven Studie auf Basis archivierter Daten beschrieben. Anschließend finden der Entwurf und die Implementierung eines Werkzeugs statt, welches die Analyse von archivierten Webseiten ermöglicht. Nach Durchführung der Studie werden die Ergebnisse präsentiert, mittels verwandter Arbeiten evaluiert und abschließend diskutiert.

3

ENTWICKLUNG DES WEB-TRACKINGS

Zusammenfassung: Grundlage einer jeden Untersuchung ist eine intensive Betrachtung des aktuellen Stands der Forschung. Zunächst wird die Entwicklungsgeschichte des Webs und Web-Trackings näher ausgeführt. Zur Vorbereitung einer retrospektiven Analyse werden anschließend bestehende quantitative Analysen näher betrachtet. Da eine genauere Untersuchung der Technik des Web-Trackings für Kapitel 7 vorgesehen ist, werden Arbeiten zu diesen Themen auszugsweise vorgestellt.

3.1 ANFÄNGE DES WEB-TRACKINGS

Anfänge

In den Anfängen des Internets wurde das Interesse der Besucher einer Webseite auf Basis der Auswertung von Serverlogs gemessen [181]. Der Webserver protokolliert alle Zugriffe in eine Logdatei, welche die Merkmale Zeit, IP-Adresse und ggf. weitere Metainformationen beinhaltet. Mit steigendem Interesse am Internet nahm der Bedarf an besseren Analysen zu. Um diese zu erleichtern, stellte die *Urchin* Software Corporation das Werkzeug *Urchin* zur Verfügung.

Lokale Auswertung von Serverlogs

Bei einer solchen lokalen Logauswertung zeigen sich verschiedene Nachteile. Nicht jeder Webanbieter war in der Lage einen eigenen Internetserver zu betreiben, weshalb Speicherplatz extern angemietet wurde. Sofern dieser angemietet wurde, bestand möglicherweise keinen Zugriff auf diese Logginformationen. Zudem sind nicht alle Inhalte von Interesse wie die automatisch nachgeladenen Inhalte via HTTP, die keine für die Auswertung nützlichen Informationen beinhalten. Diese verursachen vielmehr ein schwierig zu handhabendes Logvolumen.

Extern eingebundene Hit-Counter

Aufgrund dieser Nachteile wurden im Amateurbereich Hit-Counter verwendet. Hit-Counter waren Einbettungen, die bei Besuch der Webseite ausgelöst wurden und zu einer Inkrementierung eines Zählers in einer Datenbank führten, was üblicherweise auch die Anzeige auf der Webseite aktualisierte. Diese konnten leicht in bestehende Webseiten integriert werden und neben der reinen Zugriffsstatistik zusätzlich weitere Daten aufnehmen. In Abbildung 3.1 ist ein Beispiel für einen solchen Hit-Counter[1] abgebildet.

Urchin-Modi

Um eine vergleichbare Funktionalität zu bieten, stellte die Urchin Software zwei Modi zur Verfügung: einen zur beschriebenen Auswertung von Serverlogs und einen weiteren, der über Einbettungen auf Webseiten (Page-

1 Eingebettet auf der Webseite khemorex-klinzhai.de, im Archiv abrufbar: https://web.archive.org/web/20000511050419/http://www.khemorex-klinzhai.de:80/, abgerufen am 27.02.2018.

Abbildung 3.1: Klickcounter von digits.com eingebettet auf der Webseite khemorex-klinzhai.de aus dem Jahr 2000.

Tagging[2]) umgesetzt wird. Diese ließen sich miteinander kombinieren (hybrider Modus).

Sprung ins Web-Tracking

Während die lokale Auswertung von Serverlogs Web-Analytics zugesprochen werden kann, stellen solche Einbettungen den Sprung ins Web-Tracking dar. Die Nutzung dieser Umsetzungsform, später durch JavaScript erweitert, erlauben neben umfassenden Datenauswertungen nun auch Analysen der Browserverläufe. Die erhobenen Daten können anschließend für vielfältige Zwecke eingesetzt werden.

Ende von Urchin

Zu beachten ist, dass diese Analysemöglichkeiten auch von anderen Unternehmen angeboten wurden. Urchin wurde als Beispiel gewählt, weil sich kein anderes so erfolgreich entwickelt hat: Im Jahr 2005 wurde die Urchin Software Cooperation von Google Inc. übernommen und ist heute als Google-Analytics bekannt. Die ursprüngliche Urchin Software wurde noch weitere 7 Jahre eingesetzt und schließlich im Jahr 2012 eingestellt[3]. Google-Analytics setzt vollständig auf die Einbettung als Webseitenkomponente, ohne dass dabei ein Eigenbetrieb möglich ist.

3.2 HISTORISCHE ENTWICKLUNG

Zustandslosigkeit von HTTP

Die technische Entwicklung beginnt mit einer wesentlichen Eigenschaft bzw. Designentscheidung von HTTP: Wie in der Dokumentation von HTTP 1.0 in RFC 1945 vom Mai 1996 [19] entnommen werden kann, wird nach jeder Antwort des Servers die Verbindung zum Client (Browser) geschlossen:

> „[...] current practice requires that the connection be established by the client prior to each request and closed by the server after sending the response."
> Quelle: [19]

Protokolleigenschaften

Aus diesem Grund wird HTTP auch als zustandsloses Protokoll angesehen[4]. Eine Sitzung, die mehrere Webseitenaufrufe umfasst, muss durch eine zusätzliche Methodik (z. B. auf Basis parametrisierter Links) umgesetzt werden. Die darunter liegenden Übertragungsschichten wie Transport- oder

2 https://brianclifton.com/blog/2007/10/07/hosted-v-software-v-hybrid-tools/, abgerufen am 27.02.2018.

3 https://analytics.googleblog.com/2012/01/end-of-era-for-urchin-software.html, abgerufen am 27.02.2018.

4 „The Hypertext Transfer Protocol (HTTP) is a stateless application-level protocol for distributed, collaborative, hypertext information systems." nach RFC 7230 [56].

Vermittlungsschicht (üblicherweise TCP/IP) eignen sich nur bedingt zu diesem Zweck: Die Zuordnung einer Sitzung auf Basis der IP-Adresse ist insbesondere bei Einsatz von „Network Address Translation“ oder Proxys fehleranfällig. Die Weiterentwicklung zu HTTP 1.1 [55, 121] im Jahr 1999 ermöglichte, eine bereits etablierte Verbindung zum Web-Server für weitere Anfragen offen zu halten (Keep-alive). Dies diente zwar der Minimierung von Protokolloverhead der darunterliegenden Übertragungsschichten, aber nicht der Etablierung von Sitzungen. In der aktuellen Version vom Mai 2015 (Stand: Oktober 2017) wurden mit HTTP/2 [15] Optimierungen der Performanz eingeführt. Beispiele sind neue Kompressionsmöglichkeiten und Datenübertragungen, die vom Server initiiert werden.

Einführung von Cookies

Mit den von Lou Montulli im Jahr 1994 entwickelten Cookies sind Sitzungen auf Basis von HTTP möglich. Als Bestandteil der RFC 2109 aus dem Jahr 1997 [99] dienen HTTP-Cookies als Träger von Zustandsinformationen. Diese Technik wurde bis zur aktuellen Version RFC 2965 (Stand: 2016) aus dem Jahr 2011 [12] weiterentwickelt. Um zwischen Benutzer (User-Agent) und Anbieter (Webserver) eine Sitzung zu etablieren, die mehrere unabhängige Client/Server-Verbindungen überdauern kann, müssen diese untereinander in einen gemeinsamen Kontext gebracht werden. Aus diesem Grund werden in RFC 2109 [99] die HTTP-Header-Felder `Cookie` und `Set-Cookie` eingeführt. Eine bereits zuvor implementierte Cookie-Lösung vom Netscape Browser war mit dieser kompatibel. Das Protokoll sieht vor, dass eine vom Server mit `Set-Cookie` angegebene Zeichenkette in nachfolgenden Anfragen des Clients im Header mittels `Cookie` Feld übermittelt. Es muss beachtet werden, dass auf diese Weise eine Identifizierung eines Clients über einen längeren (vom Server festgelegten) Zeitraum möglich ist. Dies geschieht üblicherweise ohne Kenntnisnahme durch den Nutzer oder dessen Einwilligung.

Aufbau von Cookies

Nach RFC 6265 [12] besteht ein Cookie aus einer Liste von Zuweisungen, die einen Schlüssel (`cookie-name`) mit einer Zeichenkette (`cookie-value`) verbinden. Darüber hinaus sieht die Spezifikation weitere Attribute vor. So kann die maximale Lebensdauer eines Cookies über die Attribute `Max-Age` und/oder `Expires` spezifiziert werden, wobei `Max-Age` Vorrang genießt. Das Cookie-Objekt beinhaltet zudem Informationen über die Domain, von der es gesetzt wurde und Secure-Flag schützt das Cookie vor einer unverschlüsselten Übertragung. Bereits in der Spezifikation wird darauf hingewiesen, dass nicht davon auszugehen ist, dass alle User-Agents sämtliche Attribute implementieren.

Isolierung von Webseiteninformationen

Cookies ermöglichten Besucher von Webseiten über einen längeren Zeitraum wiederzuerkennen. Diese Cookies zu erhalten bzw. vom User-Agent übermittelt zu bekommen, ist nur dem Webserver gestattet, der das Cookie gesetzt hat. Bereits in der ersten Version der Cookie-Spezifikation [99] aus dem Jahr 1997 wurde im Abschnitt 8 „Security Considerations“ etwas angeregt, was derzeit als Same-Origin-Policy bekannt ist:

> „[...] For example, a malicious server could embed cookie information for host a.com in a URI for a CGI on host b.com. User agent implementors are strongly encouraged to prevent

this sort of exchange whenever possible."
Quelle: [99]

Dieser Hinweis empfiehlt den Entwicklern von User-Agents, Cookies unterschiedlicher Domains strikt voneinander zu trennen, da Cookie-Informationen sonst zur Übernahme etablierter HTTP-Sitzungen durch Dritte (Session-Hijacking) genutzt werden können.

Same-Origin

Die Same-Origin-Policy dient der Privacy und verhindert, Benutzer auf direktem Weg über mehrere Webseiten hinweg zu verfolgen. Der Betreiber der Domain B kann nicht direkt auf Sitzungsinformationen von A zugreifen. So erhält B keine Informationen darüber, ob der Besucher vorher die Domain A besucht hat. Müssen diese Verlaufsinformationen erhoben werden, ist dies nur auf zwei Wegen möglich:

1. die Anbieter (hier: A und B) erheben diese Daten getrennt, verarbeiten die Informationen (z. B. Serverlogs vom Web-Server) und tauschen diese untereinander aus,
2. der Nutzer teilt der aufgerufenen Webseite (hier: B) mit, welche Webseiten er zuvor (hier: A) aufgerufen hat.

server- vs. clientbasiert

Da bei Variante 1 die Aktivität allein von den beteiligten Servern ausgeht, wird diese in der Literatur als serverbasierte Analyse bezeichnet. Hingegen wird bei Variante 2 der User-Agent selbst aktiv und ist als clientbasiertes Verfahren zu verstehen [181]. Während letztere von außen messbar sind, gibt es derzeit keine Studien über das Ausmaß von serverbasierten Verfahren.

Informationsweitergabe durch den Browser selbst

Messbar ist hingegen die Weitergabe an Informationen, die der User-Agent selbst bewirkt. Ein Beispiel hierfür ist der bereits in HTTP 1.0 [19] enthaltene HTTP-Header `Referer`. Bei einer HTTP-Anfrage enthält dieses Feld die Adresse, von welcher die Anfrage hervorgerufen wurde. Folgt ein Besucher einem Link von der Webseite A auf Webseite B, enthält die Anfrage an den Webserver von B die Ursprungsadresse, die URI von A. Damit ist B in der Lage zu messen, wie viele Besucher den Webauftritt über einen Verweis von A erreicht haben. Der Referrer wurde zwar in seiner Form bis zum heutigen HTTP/2 Standard beibehalten, jedoch 1996 schon als möglicherweise problematisch eingestuft. So enthält RFC [19, S. 45] einen entsprechenden Hinweis:

Weitergabe privater Informationen über den HTTP-Header.

„ Note: Because the source of a link may be private information or may reveal an otherwise private information source, it is strongly recommended that the user be able to select whether or not the Referer field is sent. [...]"
Quelle: [19, S. 45]

Wie bereits beschrieben, können parametrisierte Links für eine solche Informationsweitergabe genutzt werden. So wäre es A möglich, die URI zu B so anzupassen, dass B über die Herkunft des Benutzers von der Webseite A informiert ist; der Path oder Query Teil einer URI eignet sich für diesen Zweck. Dies wäre jedoch entweder eine spezielle Absprache einer geschlossenen Gruppe von Webseiten oder müsste für den allgemeinen Fall (z. B.

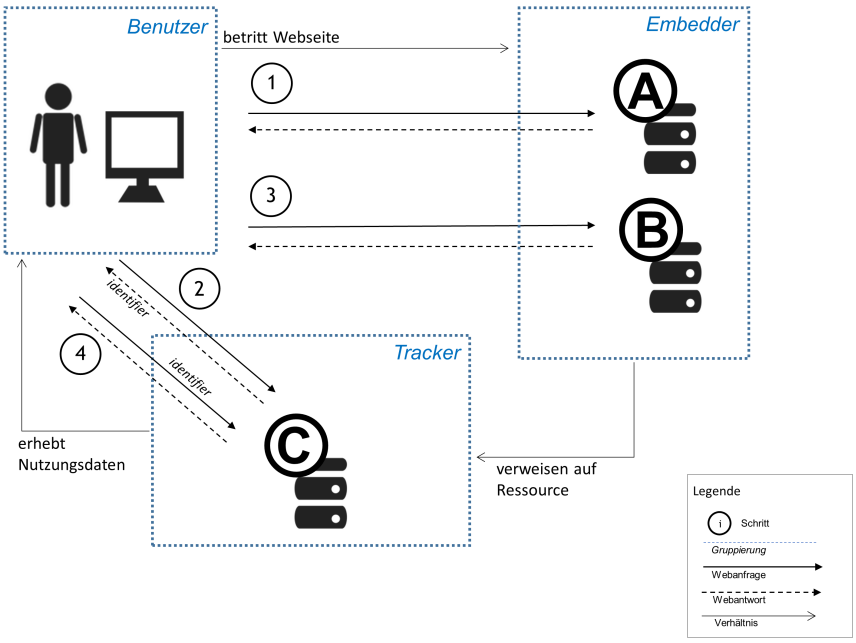

Abbildung 3.2: Darstellung von Third-Party Web-Tracking

mittels RFC) spezifiziert sein. Die Auswertung des Referrers sowie die Verwendung solcher parametrisierten Links liefern ausschließlich Informationen über den direkten Vorgänger. Sie erfordern ggf. komplexe Änderungen der Webseiten, da alle nach außen führenden Links angepasst sein müssen.

Third-Party Web-Tracking

Diese und weitere Gründe mögen dazu geführt haben, dass sich Third-Party Tracking etabliert hat[5]. Anstelle einer Absprache zwischen A und B binden beide Webseiten eine Ressource der Drittpartei C ein. Dies bewirkte, dass der Browser bei Aufruf der Seite des Anbieters A eine weitere Verbindung zu C erzeugt. Wechselt der Benutzer zu einem späteren Zeitpunkt zum ebenfalls teilnehmenden Anbieter B, findet erneut eine Kommunikation mit C statt. Auf diese Weise wurde der Tracker C über die Aufrufe der Webseiten A und B informiert. Diese Informationen können von C für verschiedene Zwecke (z. B. Werbung) genutzt werden.

Zur Entstehung von Web-Bugs

Die Art der Einbindung ist vom gewählten Verfahren abhängig. In den Anfängen des Word Wide Web handelte es sich üblicherweise um eine 1x1-Pixel große Bilddatei, die als Web-Bug oder Web-Beacon bekannt ist und von Alsaid et al. [6] sowie Martin et al. [114] genauer definiert wurde. Nach Bennett [17] wurden diese anfänglich auch in E-Mails und Dokumenten eingesetzt.

Third-Party Anfragen

Diese Vorgehensweise ist für die Anbieter mit wenig Aufwand verbunden. Die Erstpartei (First Party, hier: A) muss lediglich eine Einbettung vornehmen, wodurch der Drittpartei (Third Party, hier: C) die Verfolgung eines Besuchers über andere Anbieter hinweg möglich ist. In der Literatur finden

5 Die Beschreibung der Entwicklung von Third-Party Tracking ist ein Bestandteil dieser Arbeit.

sich abweichende Aussagen darüber, ob es sich bei der Zweitpartei (Second Party) um eine andere Webseite (B) oder um den Besucher selbst handelt. Die einbindende Partei (hier: A) wird als Embedder bezeichnet und die Drittpartei (hier: C) als Tracker. Diese Rollenverteilung wird in Abbildung 3.2 verdeutlicht.

Technische Ziele von Trackingverfahren

In Bezug auf den Besucher muss der Tracker zwei Ziele verfolgen:

1. Eine möglichst umfangreiche Erhebung der Nutzer- bzw. Nutzungsinformationen und
2. die Möglichkeit den Besucher auch langfristig verfolgen bzw. einer Person zuordnen zu können.

Datenerhebung

Das erste Ziel kann über die Form der Einbettung vom Tracker beeinflusst werden. Es ist möglich, anstelle einer Bilddatei einen aktiven Inhalt auszuliefern, welcher auf dem System des Besuchers ausgeführt wird. Ein Beispiel hierfür ist die Verwendung von JavaScript, welche eine noch umfangreichere Erfassung des Benutzerverhaltens ermöglicht. So lassen sich beispielsweise die exakte Verweildauer ermitteln, Mausbewegungen analysieren und weitere Systeminformationen erfassen. Dieses Vorgehen kann als Tiefenanalyse betrachtet werden, da sie dem Tracker tiefgehende Einblicke in die Verhaltensweise des Besuchers ermöglichen.

Robustheit

Das zweite Ziel stellt die Robustheit des Trackings sicher. Im Laufe der Jahre wurden Methoden entwickelt, welche die Wiedererkennung eines Besuchers ermöglichen. Dieses Ziel kann durch Cookies verfolgt werden. Werden diese gelöscht, ist eine Wiedererkennung nicht länger möglich. Aus diesem Grund haben sich weitere Verfahren etabliert, die als Ersatz von gelöschten Cookies dienen können. Um dies umzusetzen, wird auf andere Speicherorte zurückgegriffen (z. B. Cache) oder auf Anwendungen, welche die Browserfunktionalität erweitern (z. B. Adobe Flash). Auf diese Weise können Cookieinformationen, die durch den Benutzer gelöscht wurden, wiederhergestellt werden.

3.3 Untersuchungen zur Technik von Web-Tracking

Überblick zu wissenschaftlichen Arbeiten

Im Laufe der Entstehungsgeschichte sind Publikationen entstanden, die ein neues Web-Tracking-Verfahren beschreiben, neue Schutzmaßnahmen aufzeigen oder quantitative Analysen vornehmen. Die quantiativen Studien werden in Abschnitt 3.4 gesondert berücksichtigt. In diesem Abschnitt wird ein erster Überblick bereitgestellt, wie das Thema Web-Tracking akademisch behandelt wurde. Eine genauere Betrachtung und Sortierung bzgl. der Verfahren und Schutzmaßnahmen wird in Kapitel 7 vorgenommen.

3.3.1 *Verfahren*

Überblick zu Verfahren

- Mayer und Mitchell [115] stellten im Jahr 2012 in einem Übersichtspapier sowohl aktuelle Verfahren als auch Schutzmaßnahmen zusammen und nahmen einen Strukturierungsversuch vor.

- Speicherverfahren, hier die Verwendung von Flash, eigenen sich als Cookie-Ersatz und wurden von McDonald und Cranor im Jahr 2011 näher untersucht [117].
- Neben Einsatz von Flash als neue Trackingtechnologie betrachten Ayenson et al. [9] im Jahr 2011 auch die Verwendung von E-Tags und HTML5 als Ersatz von Cookies.
- Im Jahr 2011 demonstrieren Boda et al. [24] Fingerprint-Verfahren, die eine Cross-Browser-Wiedererkennung ermöglichen.
- Bujlow et al. [28] stellen in einem Übersichtspapier aus dem Jahr 2015 Trackingverfahren und Schutzmaßnahmen zusammen.
- Nikiforakis et al. [133] und Mulazzani et al. [126] demonstrieren verschiedene Fingerprint-Verfahren, erklären deren Funktionsweise und geben einen Ausblick auf den kommerziellen Einsatz.
- Fifield und Egelman [58] stellen in einer Veröffentlichung von 2015 ein spezielles Verfahren vor, welches auf Basis installierter Schriftarten die Erstellung eines Fingerabdrucks (des Systems) ermöglicht.
- Yen et al. [205] zeigen, wie ein Benutzer allein durch Informationen identifiziert werden kann, die bei einem Webseitenabruf üblicherweise übermittelt werden und grenzen auf diese Weise aktives von passivem Fingerprinting ab. Die Arbeit wird insbesondere in Abschnitt 8.2.5 genauer betrachtet.
- Krishnamurthy et al. [97] präsentieren verschiedene Typen von Trackern und in welcher Weise Nutzerinformationen weitergegeben werden.
- Olejnik et al. [138] nutzen Lade- bzw. Entladeinformationen der Batterie (eines Notebooks oder Mobilgerätes, zugreifbar über HTML5), um einen Nutzer zu identifizieren.
- Janc und Olejnik [88] zeigen, wie der Browserverlauf durch einen CSS-basierten Angriff ausgelesen werden kann. Wird dies vom Browser unterbunden, hilft möglicherweise ein Seitenkanalangriff, wie er von Weinberg et al. [197] beschrieben wird.
- Krishnan et al. [98] zeigen, wie durch aktiviertes DNS Prefetching[6] eine ungewollte Informationsweitergabe bewirkt wird.
- Panchenko et al. [140] demonstrieren einen Fingerprinting-Angriff, der insbesondere auf Nutzer in anonymisierten Netzwerken (TOR, JAP) abzielt.

3.3.2 *Schutzmaßnahmen*

Überblick zu Schutzmaßnahmen

- Roesner et al. [153] entwickelten 2012 ein Klassifikationsschema, auf dessen Basis verschiedene Schutzkonzepte erarbeitet wurden.

6 Dabei werden Hostadressen, die auf einer Webseite durch Links hinterlegt sind, bereits während dem Ladevorgang aufgelöst.

- Acar et al. [1] stellten 2013 eine Methodik zur Erkennung von Fingerprinting-Verfahren vor und entwickelten darauf basierend die Firefox-Erweiterung „FPDetective“.
- Wu et al. [203] verwenden Verfahren zum maschinellen Lernen zur Erkennung von Third Party Trackern und erreichen dabei eine Trefferquote von 97,7 %.
- Malandrino et al. entwickelten die Browser-Erweiterung „NoTrace“ und grenzen dessen Funktionsumfang zu anderen (zu dieser Zeit) marktüblichen Erweiterungen in einer Veröffentlichung [112] von 2013 ab.
- Jackson et al. [87] zeigen, weshalb die Same-Origin-Policy, die bei Cookies bereits Anwendung findet, auch auf andere Teile des Browsers (z.B. den Cache) ausgeweitet werden sollte. Die Isolierung von Webseiten verfolgen Pan et al. [139] beim Design ihres „Trackingfree“-Browsers. So auch Chen et al [32], die eine möglichst starke Trennung der Daten anstreben.
- Backes et al. [10] schlagen mit „ObliviAd“ ein Hardwaremodul vor, welches die Einblendung personalisierter Werbung ermöglicht, ohne dass personenbezogene Informationen an die Werbeindustrie weitergegeben werden.
- Balebako et al. [11] messen die Effektivität verschiedener Schutzmaßnahmen und wie effektiv personalisierte Werbung darüber unterbunden werden kann.
- Fredrikson und Livshits [59] entwickeln mit „RePriv“ ein Protokoll, um die Weitergabe von personenbezogenen Informationen transparenter zu gestalten. Es verfolgt damit einen vergleichbaren Ansatz wie P3P.
- Leon et al. [101] führen einen Vergleich von neun Schutzmaßnahmen sowie eine Befragung von 45 Personen bezüglich deren Umgang mit solchen Schutzmaßnahmen durch.
- Li et. al. [105] demonstrieren mit „TrackAdvisor“ ein auf maschinellem Lernen basierende Schutzmaßnahme gegen Third-Party Tracking.

3.4 Untersuchungen zur Ausbreitung von Web-Tracking

Ausbreitung des Web-Trackings

Wie im vorherigen Abschnitt beschrieben, existieren technische Weiterentwicklungen, welche den Einsatz von Trackingmaßnahmen ermöglichen, vereinfachen oder begünstigen. Damit ist noch keine Aussage getroffen, wie stark sich dessen Verwendung verändert hat.

Grundsätzlich wird von einer Zunahme von Web-Tracking ausgegangen [177, 96]. Ein Vergleich von bestehenden Veröffentlichungen [44, 153, 115, 161, 2] zeigen eine Zunahme, dienen aufgrund der unterschiedlichen Erfassungsmethodik und Erfassungszeiträume jedoch nicht dazu ein Gesamtbild aufzuzeigen.

Die Publikationen können in drei Klassen unterteilt werden. Die erste Klasse sind allgemeine Untersuchungen, die neben einer Übersicht zum aktuellen Wissensstand zu Web-Tracking eine Erhebung durchführen. Die zweite Klasse umfasst Publikationen, die spezielle Zielgruppen fokussieren, z. B. die Ausbreitung sozialer Netzwerke im Web. Die dritte Klasse beschreibt spezielle technische Verfahren (z. B. der Einsatz von Adobe Flash) zur Durchführung von Web-Tracking, welche anschließend im Web gemessen werden.

ALLGEMEINE UNTERSUCHUNGEN Bei dieser Klasse handelt es sich um Analysen, die sich mit der allgemeinen Ausbreitung beschäftigen, ohne eine (zu) spezielle Einschränkung der Webseiten oder der Technik vorzunehmen. Diese Art der Analyse ist häufig in Übersichtspapieren zu finden, um dem Leser einen Überblick bzgl. des Ausmaßes von Web-Tracking zu geben.

- Von Krishnamurthy et al. [96] wurde im Jahr 2009 eine Langzeitstudie veröffentlicht, welche im Vergleich zu den anderen Arbeiten einen größeren Zeitraum abdeckt. Dabei wurden 5 Messungen von ca. 1200 Webseiten durchgeführt, die zwischen Oktober 2005 und September 2008 verteilt wurden.
- Libert [107] führte im Mai 2014 eine vergleichsweise umfassende Analyse von 1 000 000 Webseiten bzgl. Web-Tracking durch. Dabei handelt es sich nur um eine Momentaufnahme des jeweiligen Jahres.
- Englehardt und Narayanan [48] stellen die Umgebung OpenWPM für automatische Analysen vor und zeigen die Ergebnisse der Messung von 1 000 000 Webseiten.

SPEZIELLE GRUPPEN In weiteren Veröffentlichungen werden spezielle Arten von Webseiten, Trackern oder Regionen betrachtet.

- Von Chaabane et al. [30] wurde 2011 eine Auswertung der Alexa Top 10 000 durchgeführt, welche das Ausmaß der Einbettungen so genannter Social Media Plugins betrachtet. Dabei wird insbesondere analysiert, welche Möglichkeiten soziale Netzwerke zur Nutzerverfolgung besitzen.
- Von Falahrastegar et al. [52] wurde 2014 eine quantitative Analyse von Web-Tracking verschiedener Regionen vorgestellt. Dabei wurden ausgehend von 7 unterschiedlichen Herkunftsorten 500 Webseiten analysiert und ihre Unterschiede betrachtet.

SPEZIELLE TECHNIKEN In einigen Veröffentlichungen wurden neue Trackingverfahren vorgestellt und bezüglich dieser speziellen Technik eine Ausbreitungserhebung durchgeführt.

- McDonald und Cranor [117] analysieren eine spezielle Art von Flash-basiertem Web-Tracking und betrachten zu diesem Zweck 100 häufig besuchte und 500 zufällig gewählte Webseiten. Einen ähnlich speziellen Fokus haben Ayenson et al. [9] und analysieren 1200 Webseiten auf Flash- und HTML5-basierten Cookies.

- Jang et al. [89] führt eine genauere Analyse von JavaScript-Aktivitäten auf 50 000 Webseiten durch, welche auch Web-Tracking-Verfahren einschließen.
- Bau et al. [14] führt eine Analyse von 32 000 Webseiten als Evaluierung für einen Algorithmus auf Basis maschinellem Lernens durch.
- Nikiforakis et al. [133] analysiert drei als populär eingestufte Fingerprint-Verfahren und evaluiert die Erkennung an 10 000 Webseiten mit jeweils 20 Unterseiten.
- Fruchter et al. [60] untersucht Tracking- und Webseitenverhalten sofern IP-Adressen aus verschiedenen Regionen eingesetzt werden. Dies wird mit 250 Webseiten unter Verwendung von je 4 verschiedenen IP-Adressen evaluiert.
- Mikians et al. [119] untersucht Preisdiskriminierung auf Basis der IP-Adresse (IP Location). Dafür werden Webseiten von 200 Anbietern über einen Zeitraum von 20 Tagen mehrmals abgerufen.
- Stopcynski und Zugelder [175] evaluieren eine Schutzmaßnahme auf Basis von Speicherisolation anhand 40 000 Webseiten mit jeweils 50 Unterseiten.
- Acar et al. [2] betrachtet insbesondere ein Fingerprinting-Verfahren und analysiert diesbezüglich 100 000 Webseiten.
- Schelter und Kunegis [159] führten 2015 eine sehr umfassende Analyse von ca. 3,5 Milliarden HTML Webseiten durch. Bei der Interpretation der Daten muss jedoch die besondere technische Umsetzung berücksichtigt werden. So wurde der HTML-Quelltext statisch nach *script*, *image* und *iframe* HTML-Elementen durchsucht sowie JavaScript-Quelltext interpretiert ohne ihn auszuführen. Die reine textuelle Suche der von CommonCrawl[7] zur Verfügung gestellten Daten unterscheidet sich das Verfahren von anderen Publikationen wie Libert [107] oder Englehardt und Narayanan [48], die einen erweiterten oder modifizierten Browser einsetzen.

Wichtige quantitative Studien

Allein Krishnamurthy et al. [96] führen eine Studie durch, die den Einsatz von Web-Tracking über einen längeren Zeitraum betrachtet und vergleicht. Ansonsten werden nur einmalige Erfassungen durchgeführt oder sehr kurze Zeitabstände (wenige Tage) gewählt.

Hinweis: Lerner et.al. [103] stellten im August 2016 eine Studie zur Entwicklung von Web-Tracking Technologien vor, die eine Auswertung des Internet Archives vornimmt. Es ist zu beachten, dass diese Veröffentlichung erst nach der Publikation der Ergebnisse dieser Dissertation [191] (Februar 2016) erfolgte. Eine entsprechende Abgrenzung wird in Abschnitt 6.3.2 vorgenommen und mit den Ergebnissen dieser Arbeit verglichen.

7 http://commoncrawl.org/, abgerufen am 27.02.2018.

4

DESIGN DER RETROSPEKTIVEN STUDIE

Zusammenfassung: In diesem Kapitel wird das Design einer retrospektiven Studie beschrieben, um die Entwicklung von Web-Tracking zu erfassen. Nach einer Einleitung werden die Untersuchungsmethodik, verwandte Arbeiten sowie verfügbare Datenquellen näher betrachtet. Ziel ist die Festlegung von Anforderungen, die in Entwurf und Implementierung eines dafür vorgesehenen Werkzeugs berücksichtigt werden müssen.

4.1 EINLEITUNG ZUR RETROSPEKTIVEN ANALYSE

Entwicklung des Webs

Durch Änderungen von Webseitenbetreibern selbst, als auch durch Interaktion von Benutzern mit dynamisch generierten Webseiten unterliegt das World Wide Web einem permanenten Wandel [35]. Der Aufbau von Webseiten hat sich in den letzten 20 Jahren auf vielfältige Weise weiterentwickelt: Neue Technologien, die auf leistungsstärkerer Hardware ausgeführt werden, ermöglichen die Übertragung und Darstellung von Bildern, Videos und Anwendungen [72, 50]. Auch der Netzausbau ermöglicht einer wachsenden Personengruppe einen leistungsfähigeren Internetanschluss[1], was sich ebenfalls auf die Gestaltung von Webseiten auswirkt: Kahle [92] zeigt eine fast Verdreifachung der Größe durchschnittlicher Webseiten von ca. 700 Kilobyte auf 2 Megabyte allein zwischen den Jahren 2010 und 2015.

Verlust von Daten im Web

Neue Versionen von Webauftritten ersetzen die vorherigen und sind nicht länger verfügbar. Sofern diese nicht durch spezielle Versionsverwaltungssysteme gesichert wurden, sind alte Stände für die Nachwelt verloren. Dieser Fakt wurde bereits Mitte der 90er Jahre erkannt, in denen die ersten Web-Archive-Projekte gestartet sind. Diese haben sich die Erhaltung von Webinhalten zur Aufgabe gemacht, indem der aktuelle Stand von ausgewählten Webseiten kontinuierlich erfasst und gespeichert werden.

Analysen von Archivdaten

Wie in Abschnitt 3.4 beschrieben wurde, sind nur vereinzelte quantitative Analysen zur Entwicklung von Web-Tracking verfügbar. Diese umfassen nur kurze Zeitspannen und geben keinen ausreichenden Gesamtüberblick über die Ausbreitung von Web-Tracking. Es stellt sich die Frage, ob archivierte Daten genügen, um rückblickende Untersuchungen bzgl. dieser Fragestellung durchzuführen.

Veränderung von Web-Tracking

Ziel ist es, die durch Web-Tracking entstandenen Veränderungen des Internets rückblickend zu erfassen und auszuwerten. In diesem Kapitel werden das Konzept einer solchen Analyseform näher erläutert und Anforderungen an die Implementierung erarbeitet.

1 17. TK-Marktanalyse Deutschland 2015, Seite 20, http://www.vatm.de, abgerufen am 11.09.2016.

Teile dieser Arbeit wurden in den von mir verfassten Veröffentlichungen [191] und [192] publiziert:

- Bei [191] handelt es sich um die Präsentation der Methodik und eine anschließende Auswertung von 10 000 Webseiten. Die Veröffentlichung wurde am 21. Oktober 2015 eingereicht und am 21. Februar 2016 in Rom (Italien) vorgestellt.
- Nach Präsentation der Ergebnisse forderten die Konferenzveranstalter eine erweiterte Version an, welche im April 2016 eingereicht und publiziert [192] wurde.

Die folgenden Ergebnisse entsprechen der erweiterten Version [192], wobei Methodik und Implementierung identisch mit [191] sind.

4.2 Methodik

4.2.1 *Forschungsfragen*

Problemstellung und Fragen

Die in Abschnitt 3.4 vorgestellten Arbeiten sind in ihrer Vorgehensweise sehr verschieden und bieten keine direkten Vergleichsmöglichkeiten. Darüber hinaus erfassen nur wenige Studien einen längeren Zeitraum und können aufgrund ihres Alters nicht länger als aktueller Stand betrachtet werden. Ein Gesamtbild über die Entwicklung von Web-Tracking kann neben der Informatik auch anderen Disziplinen als Ausgangsbasis für gesellschaftliche Theorien dienen.

> Forschungsfrage RQ-1 In welcher Weise hat sich Web-Tracking in den vergangenen Jahren ausgebreitet?
>
> RQ-1.1 Welche Daten stehen zur Generierung eines Gesamtbildes zur Verfügung?
>
> RQ-1.2 Mit welcher Technik kann eine solche Entwicklung erfasst werden?

Die abgeleiteten Forschungsfragen stellen den technologischen Aspekt in den Vordergrund. Hauptziel ist das Sammeln und Bereitstellen der Daten, die unter Berücksichtigung des jeweiligen Forschungsgegenstandes in verschiedene Richtungen vielfältig interpretiert werden können. Damit wird zudem eine Aussage über die Zu- oder Abnahme von Web-Tracking ermöglicht.

4.2.2 *Forschungsmethodik*

Der DSR-Zyklus

Die Forschungsfrage soll unter Anwendung von Design Science Research (Vaishnavi und Kuechler [187]) nachgegangen werden. Die Methodik wurde bereits in Abschnitt 1.3 vorgestellt. Das Modell umfasst die fünf Phasen:

- Awareness of Problem,
- Suggestion,

- Development,
- Evaluation und
- Conclusion

die in diesem Abschnitt konkretisiert werden.

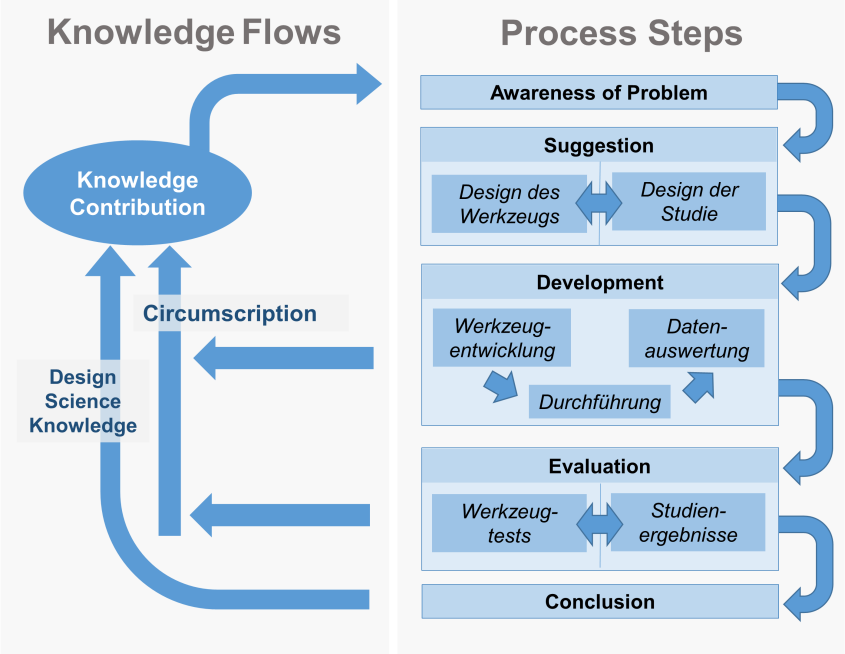

Abbildung 4.1: Konkretisierung des DSR-Zyklus', vorgestellt von Vaishnavi und Kuechler [187]. Studienbezogene Bestandteile sind deutsch und kursiv ausgezeichnet.

Awareness of Problem

Problemanalyse

Zum Analysieren des Problems wurde eine Betrachtung der bestehenden quantitativen Analysen in Abschnitt 3.4 durchgeführt.

Es muss hinterfragt werden, ob eine rückwirkende Analyse auf Basis der vorliegenden Daten möglich ist und wie diese umgesetzt bzw. durchgeführt werden kann. Weil das Studien- und Werkzeugdesign ineinandergreifen, muss sich das Werkzeug der Studie nach den verfügbaren Daten richten und die erwünschten Messungen nach den Möglichkeiten des Werkzeugs. Eine strikte Trennung ist aufgrund der starken Verbundenheit dieser Fragen nicht zielführend.

Suggestion

Konzeptionsphase

In dieser kreativen Phase wird ein Lösungskonzept (Suggestion) erarbeitet.

DESIGN DER STUDIE. Der Lösungsfindungsprozess besteht aus den folgenden drei Schritten:

1. Im ersten Schritt müssen alle zur Verfügung stehenden Daten gesichtet werden. Zu klären ist, welche Daten verfügbar sind, wie diese aufgebaut sind, ob diese maschinell verarbeitet werden können und ob diese zur Beantwortung der Forschungsfrage geeignet sind. Diese Betrachtung wird in Abschnitt 4.3 durchgeführt.
2. Nach Auswahl einer oder mehrerer Datenquelle werden diese im zweiten Schritt näher analysiert. Dabei ist zu erwarten, dass sich die Art der Messung auf archivierten Daten von den üblicherweise durchgeführten quantitativen Studien im Live Web unterscheidet.
3. Im dritten Schritt werden die Anforderungen an die Implementierung für das Messwerkzeug formuliert.

DESIGN DES WERKZEUGS. Unter Berücksichtigung dieser Anforderungen muss eine technische Lösung erarbeitet werden:

1. Zunächst werden die Lösungen verwandter Arbeiten sowie bestehende quantitative Analysen betrachtet.
2. Die gefundenen Lösungen werden in technische Äquivalenzklassen unterteilt. Ziel ist das finden einer Äquivalenzklasse, welche den erarbeiteten Anforderungen an das Werkzeug genügt.

Development

Entwicklungsphase

Die Entwicklung besteht aus drei Schritten:

1. Nach Auswahl der passenden Werkzeugklasse findet die Implementierung des Werkzeugs unter Berücksichtigung der aus der Designphase resultierenden Anforderungen statt (Abschnitt 5.4).
2. Im nächsten Schritt wird die Studie durchgeführt (Abschnitt 6.1). Diese umfasst das Beschreiben der Messumgebung, die Auswahl einer Testmenge sowie die Umsetzung.
3. Abschließend werden Auswertungen entwickelt und die Ergebnisse präsentiert.

Evaluation

Evaluierung der Arbeit

Die Evaluation (Abschnitt 6.3) unterteilt sich in zwei Teile: die Evaluation des Werkzeugs und die der Studienergebnisse.

EVALUATION DES WERKZEUGS. Das Werkzeug wird über eine Stichprobe manuell überprüft. Verglichen werden dabei die Ergebnisse, die das Werkzeug liefert, mit denen, die von üblichen User-Agents erzeugt werden.

EVALUATION DER STUDIE. Die Ergebnisse der Studie werden mit verfügbaren quantitativen Studien verglichen. Gemeinsamkeiten und Abweichungen werden herausgearbeitet und detailliert betrachtet und sofern möglich begründet.

Der Vergleich mit verwandten Arbeiten dient ebenfalls der Klärung, ob Anpassungen an der Studie oder am Werkzeug notwendig sind oder ob durch eine andere technische Vorgehensweise bessere Ergebnisse erwartet werden können.

Conclusion

Schlussfolgerung

Abschließend findet eine Diskussion der Gesamtergebnisse statt (Abschnitt 6.4). Ziel ist das Betrachten der Resultate und das Beantworten der gestellten Forschungsfrage RQ-1. Ein Ausblick, in welcher Weise die erhobenen Daten Verwendung finden können, wird in Abschnitt12 gegeben.

4.3 ERFASSUNG DER DATENLAGE

Aktuelle Datenlage

Studien zur Ausbreitung von Web-Tracking wurden bereits in Abschnitt 3.4 näher beschrieben. Darüber hinaus sollen weitere Datenquellen berücksichtigt werden, die zur Erzeugung des geforderten Entwicklungsbildes beitragen können.

4.3.1 *Verfügbares Datenmaterial*

Web-Tracking-Datenbanken

Betrachtung von Datenquellen

Datenbanken, die Informationen über das Ausmaß von Web-Tracking enthalten und ggf. Daten über einen größeren Zeitraum aus verschiedenen Quellen umfassen, könnten zur Beantwortung der Forschungsfrage und/oder zur Evaluierung der Ergebnisse dienen.

Innerhalb der Veröffentlichungen aus Kapitel 3 fanden sich keine Hinweise darauf, dass zentrale Datenbanken zur Erfassung von Web-Tracking aufgebaut wurden. Daher wird angenommen, dass zentrale Aufzeichnungen zur Entwicklung von Web-Tracking bislang nicht existiert.

Archivierte Webseiten

IIPC

Im Jahr 2003 wurde dazu das IIPC (International Internet Preservation Consortium) [81] gegründete: Diese Mitgliederorganisation verfolgt die Verbesserung von Werkzeugen und Standards zur Archivierung des Webs. Das IIPC listet Projekte[2], die eine solche Archivierung von Webseiten vornehmen. Bei genauerer Betrachtung zeigt sich, dass häufig nur Webseiten zu bestimmten Themen, Sprachen oder Regionen fokussiert werden wie zum Beispiel das „Íslenska vefsafnið“ (The Icelandic Web Archive). Dieses Archiv ist seit 1996 aktiv und sichert ausschließlich Webseiten von isländischen Web-Adressen. Im Folgenden werden beispielhaft weitere Projekte genannt und kurz beschrieben:

2 http://netpreserve.org/resources/member-archives, abgerufen am 14.09.2016.

UK WEB ARCHIVE. Das UK Web Archive[3] archiviert seit 2004 ausgewählte Webseiten aus dem Vereinigten Königreich. Seit April 2013 umfasst die Archivierung alle Webseiten der .uk Domain. Ziel ist es, die Inhalte und Entwicklung dieser Webseiten für spätere Generationen zur Verfügung zu stellen.

LIBRARY OF CONGRESS WEB ARCHIVE. Gemäß einer festen Policy [108] werden Webseiten ausgewählt und im »Library of Congress Web Archive«[4] seit dem Jahr 2000 archiviert: „It is part of a continuing effort by the Library to evaluate, select, collect, catalog, provide access to, and preserve digital materials for researchers today and in the future.“ Quelle: [108].

INTERNET ARCHIVE. Das Internet Archive[5], auch bekannt als *WayBack-Machine*, ist seit 1996 aktiv und gehört zu den ältesten aller genannten IIPC Mitgliedern. Es ist ebenfalls das einzige im IIPC gelistete Projekt, welches keine Selektion nach Themen, Sprachen oder Region vornimmt, sondern sich an der Liste der populärsten Webseiten (Alexa Internet, vgl. Abschnitt 4.3.1) orientiert. In den FAQs der Webseite[6] steht dazu:

> „The original idea for the Internet Archive Wayback Machine began in 1996, when the Internet Archive first began archiving the web. Now, five years later, with over 100 terabytes and a dozen web crawls completed, the Internet Archive has made the Internet Archive Wayback Machine available to the public. The Internet Archive has relied on donations of web crawls, technology, and expertise from Alexa Internet and others. [...] “ Quelle: archive.org

Bei der Erstellung einer Archivversion werden nicht alle Unterseiten erfasst. Über die Webseite des Archivanbieters kann ein neues Abbild einer Webseite beantragt werden.

Ranking von Webseiten

Übersicht von Rankingdiensten

Um neue Verfahren oder das Ausmaß von Web-Tracking quantifizieren zu können, werden innerhalb der Publikationen Messverfahren auf eine Teilmenge existierender Webseiten angewendet. Um eine solche Testmenge möglichst repräsentativ und reproduzierbar zu wählen, wird die Auswahl auf Basis eines Rankings getroffen. Die Kriterien, mit welcher die Popularität gemessen wird, ist nicht immer vollständig ersichtlich. Trotz dieser Unklarheit der Erhebung ist zu erwarten, dass sich ein großer Anteil der Internetnutzer auf diesen Webseiten bewegt und von Web-Tracking erfasst

3 http://www.webarchive.org.uk/ukwa/info/about, abgerufen am 14.09.2016.
4 https://www.loc.gov/webarchiving/, abgerufen am 14.09.2016.
5 https://archive.org/, abgerufen am 14.09.2016.
6 Archive.org, Frequently Asked Questions: https://archive.org/about/faqs.php, letzter Zugriff am 14.10.2016.

werden. Im Folgenden werden drei Datenbanken vorgestellt, die eine solche Einordnung vornehmen:

ALEXA INTERNET. Die Alexa Internet Datenbank, aktiv seit 1996 und seit 1999 Teil von Amazon.com, wird am häufigsten in den in Abschnitt 3.3 genannten Publikationen verwendet. Während einige Leistungen des Dienstes kostenpflichtig sind, wird ein kostenfreier Download von 1 Million Einträge angeboten. Nach eigener Aussage[7] wird ein Teil der Daten in gleicher Weise erhoben, wie es bei klassischem (Cross-Domain) Web-Tracking der Fall ist: Durch eine freiwillige Einbettung auf der Webseite. Ein anderer Teil wird über eine Erweiterung im Browser in Form einer Toolbar erhoben.

NETCRAFT. Seit 1995 stellt Netcraft verschiedene Dienste im Bereich Internetsicherheit, Anwendungstests und Pen-Testing zur Verfügung[8]. Neben weiteren Erhebungen bieten sie eine Liste der „Most Visited Web Sites“[9] an. Es ist zwar unklar, wie exakt die erhobenen Daten sind, jedoch bietet das Unternehmen wie bei Alexa Internet eine Toolbar als Browsererweiterung an.

QUANTCAST. Quantcast bietet verschiedene Dienstleistungen (Messungen, Werbeeinblendungen) und veröffentlicht ebenfalls eine Liste[10] von »Top Sites«. Ein Download der Daten in Textform ermöglicht eine automatisierte Verarbeitung. Zur Erhebungsform der Daten schreibt[11] das Unternehmen:

> „We collect directly measured data from the millions of web destinations and mobile apps controlled by Quantified publishers. All the data we collect is anonymous and contains no personally identifiable information (PII).“
> Quelle: quantcast.com

Weitere Datensammlungen

COMMON CRAWL. Das Common Crawl Projekt führt seit 2008 Erfassungen des Internets durch (vier pro Jahr) und stellt diese Rohdaten zum Download bereit. Seit 2013 sind diese im standardisierten WARC-Format. Um die Entwicklung des Web-Trackings aufzuzeigen, sind insbesondere die Daten der Jahre vor 2008 wichtig. Daher scheidet dieses Projekt als mögliche Datenquelle aus.

PRINTMEDIEN. Es ist unwahrscheinlich, dass Publikationen zu diesem Thema allein in gedruckter Form vorliegen. Der Vollständigkeit halber

7 http://www.alexa.com/about, abgerufen am 10.09.2016.
8 https://www.netcraft.com/about-netcraft/, abgerufen am 10.09.2016.
9 http://toolbar.netcraft.com/stats/topsites, abgerufen am 10.09.2016.
10 https://www.quantcast.com/top-sites, abgerufen am 10.09.2016.
11 https://www.quantcast.com/help-center/faqs/?prod=measure abgerufen am 10.09.2016.

wurde die Deutsche Digitale Bibliothek[12], »Deutsche Nationalbibliothek[13] und »The European Library«[14] gesichtet. Während Web-Tracking bzw. Web-Analytics in Printmedien behandelt werden [90, 180, 123, 120], zeigen sich keine Daten, die zu einer retrospektiven Studie beitragen können.

4.3.2 *Verwandte retrospektive Arbeiten*

Archiv-basierte Arbeiten

Die in Abschnitt 3.3 vorgestellten Publikationen zur Ausbreitung von Web-Tracking sind in ihrer Methodik für die angestrebte retrospektive Analyse wichtig. Darüber hinaus werden Veröffentlichungen beachtet, die bereits retrospektive Analysen auf Basis von Archivmaterial durchführen.

- Das LAWA-Projekt [171, 196, 178] (Longitudinal Analytics of Web Archive Data) ermöglicht als EU-Projekt (Project No. 258105) Wissenschaftlern verschiedener Disziplinen archivierte Inhalte strukturiert auszuwerten. Das Projekt fokussiert sich auf Webseiteninhalte, also auf die Analyse der Texte (Was) statt auf technologische Fragestellungen (Wie).
- Nikiforakis et al. stellen in [132] eine Sicherheitsanalyse von JavaScript-basierten Einbettungen auf Webseiten vor und führen eine Evaluation von 10.000 Webseiten durch. Um die zunehmende Entwicklung der Einbettung fremder Scripte aufzuzeigen, wurden archivierte Webseiten von archive.org ausgewertet.
- Soska und Christin [170] nutzen ein Webseitenarchiv für eine Vorhersage, ob eine bestimmte Webseite in (naher) Zukunft „schädlich“ (im Sinne der Systemsicherheit) sein wird.
- Hackett et al. [69] nutzen das Internet Archive, um die Entwicklung von Komplexität und Nutzbarkeit von Internetseiten zwischen den Jahren 1997 und 2002 retrospektiv und technisch zu analysieren. Dies u. a. in Hinblick auf Barrierefreiheit.

Weitere Arbeiten

Um „Web Archiving“ als eigene Forschungsdisziplin einzuführen, publizierte Michael Day im Jahr 2003 [41] eine Übersicht zu dem damaligen Stand von Archivierungsprojekten. Hockx-Yu [77] analog im Jahr 2011. Weiteres dazu auch von Costa et al. [34] und Toyoda und Kitsuregawa [184].

4.4 Auswahl und Analyse der Datenquelle

4.4.1 *Auswahl der Datenquelle*

Webseiten im Archiv

In Abschnitt 4.3 wurden Datenquellen von archivierten Webseiten vorgestellt. Es zeigte sich, dass die Auswahlkriterien der Internetarchivierungsdienste unterschiedlich sind. Für das geforderte Entwicklungsbild von Web-

12 https://www.deutsche-digitale-bibliothek.de, abgerufen am 10.09.2016.
13 https://portal.dnb.de/, abgerufen am 10.09.2016.
14 http://www.theeuropeanlibrary.org, abgerufen am 10.09.2016.

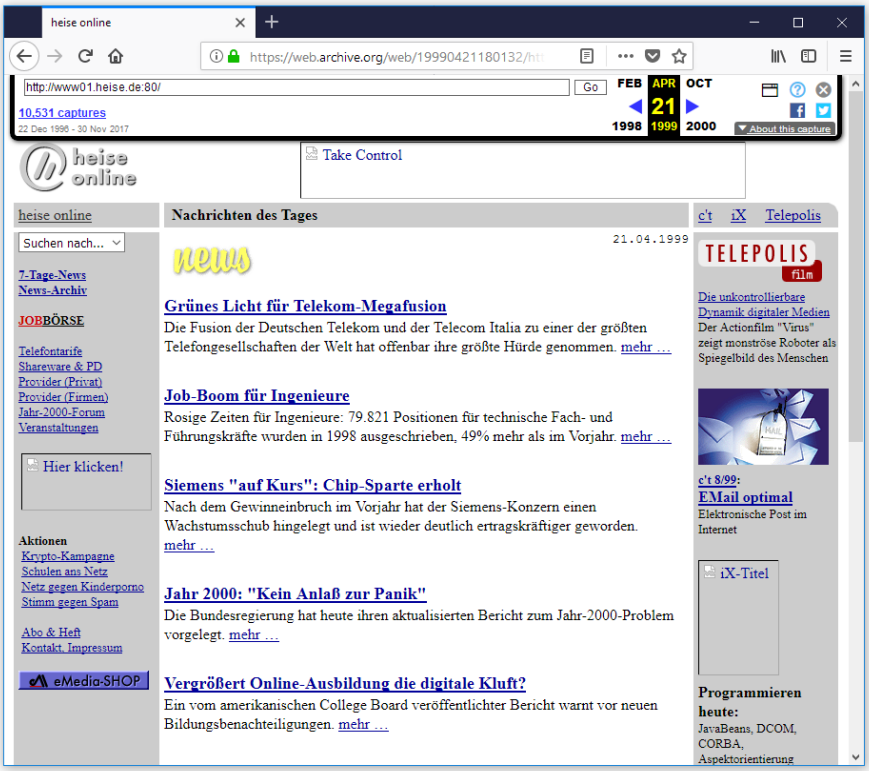

Abbildung 4.2: Internet Archive Snapshot von heise.de vom 21.04.1999 um 18:01.

Tracking sind Webseiten von Interesse, die von der „breiten Masse“ der Internetnutzer besucht werden. Solchen Webseiten hat sich das Internet Archive verschrieben, welche seine Auswahl anhand der Alexa Internet Liste trifft. Aus diesem Grund und weil es eines der ältesten Archivierungsdienste ist fällt die Wahl auf diese Datenquelle. Abbildung 4.2 zeigt ein Beispiel für eine archivierte Webseite von 1999.

Im Folgenden werden die zur Verfügung stehenden Daten vom Internet Archive analysiert.

4.4.2 *Analyse der archivierten Webseiten*

Archiv

Die archivierten Webseiten werden über einen Crawling-Algorithmus erfasst. Beim Internet Archive ist dieser Algorithmus seit 2004 unter dem Namen „Heritrix“[15] bekannt und wird u. a. von Mohr et al. wissenschaftlich behandelt [122, 164, 113].

Umfang der Archivierung

Die Archivierung verfolgt das Ziel, Quelltexte von Webseiten und damit verbundene Ressourcen zu konservieren. Grundsätzlich ist es eine Entscheidung des Archivierungsdienstes, inwieweit Unterseiten berücksichtigt bzw. bis zu welcher Tiefe diesen gefolgt werden. Es gibt vom Archivbetreiber keine klare Aussage zur Vollständigkeit der erfassten Webseiteninhalte. Es ist zu

15 http://crawler.archive.org/index.html, abgerufen am 14.10.2016.

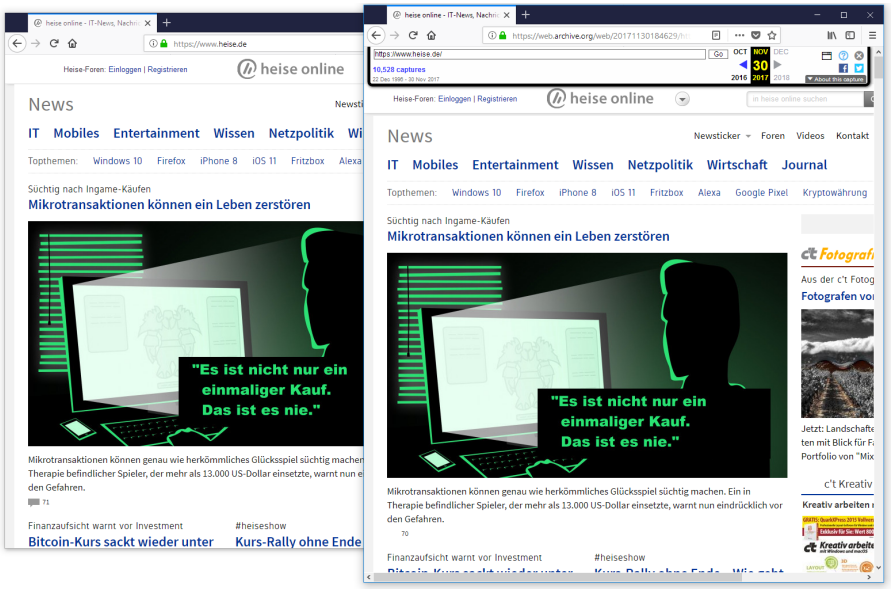

Abbildung 4.3: „Live Web“ (links) und Internet Archive Snapshot (rechts) von heise.de vom 30.11.2017 um 18:46.

erwarten, dass der Umfang stark vom Grad der Bekanntheit des Webauftritts beeinflusst wird.

Aufbau archivierter Webseiten

Aufbau der Seiten

Bei der visuellen Betrachtung archivierter Webseiten zeigen sich zunächst nur geringe Unterschiede im Vergleich zum Original im Live Web. Auffällig ist, dass der Archivbetreiber einige Steuerelemente im oberen Bereich der Webseite hinzugefügt hat, um die Navigation zwischen verschiedenen Snapshots zu erleichtern. Diese lassen sich ausblenden bzw. entfernen. In Abbildung 4.3 ist ein Vergleich zwischen einer Webseite und der archivierten Variante zu sehen.

Aufbau des Archivsystems

Vor allem bei älteren Snapshots zeigen sich Mängel: Nicht archivierte Bilder werden durch Layoutprobleme oder Dummygrafiken des Browsers sichtbar. Dies ist beispielsweise in Abbildung 4.2 zu erkennen: im linken, rechten und oberen Bereich konnten Bilder nicht geladen und angezeigt werden.

HTML-Quelltext und Einbettungen

Referenzierung

Durch den Crawling-Algorithmus wurden die Quelltexte der Webseite während des Erfassens so verändert, dass absolute und relative Pfade bei Verweisen (Hyperlinks, Einbettungen, etc.) zu ihrem archivierten Gegenstück zeigen. Wird bspw. auf der zu archivierenden Webseite der Adresse http://www.w3.org/ das Objekt `Security` referenziert (relative Pfadangabe), wird der folgende Hyperlink

```
<a href="/Security">Security</a>
```

zu einer Ressource der Archivwebseite

```
<a href="/web/<date>/http://www.w3.org/Security">Security</a>
```

geändert. Auf diese Weise wird das „Live Web“ beim Erstellen eines Snapshots mit wenigen Ausnahmen (siehe unten) zu einem homogenen „Archived Web“.

Webseiten Bestandteile

Eine Webseite besteht i.d.R. nicht nur aus dem HTML-Quelltext, sondern aus einer Reihe dazugehöriger Ressourcen: Elemente[16] wie Cascading Style Sheets (CSS), JavaScript (JS), Bilder (GIF/BMP/JPG/PNG) und eigene Schriftarten (WOF/EOT) wirken gestaltend auf die Webseite ein. Diese werden automatisch vom Browser während des Aufbaus der Webseite nachgeladen. Um einen Webauftritt möglichst authentisch zu erhalten, werden diese ebenfalls archiviert. Gleiches gilt auch für die Einbindung von anderen Webseiten durch Frames.

Zeitangaben

Beim oben beschriebenen Beispiel steht `<date>` für das jeweilige Datum des angeforderten Snapshots. Ist ein solcher nicht verfügbar, wird das nächstgelegene aufgerufen. Um Ressourcen (in Form von Speicherplatz) zu sparen, werden auch ältere Versionen von Einbettungen referenziert. Falls eine solche Sicherung nicht vorliegt sind Verweise in die Zukunft oder ins „Live Web“ möglich. Falls eine Webseite oder Ressource nicht eingelagert wurde und auch kein solcher Ersatz vorliegt, wird ein entsprechender Hinweis bzw. eine entsprechende Fehlermeldung zurückgegeben.

Zugriffe ins Live Web

Abbildung 4.4 zeigt die Abfrage des Browsers (*L1*) bezüglich einer Ressource (*R1*) an den Archivierungsserver. Die geladene Ressource (*R1*), in diesem Beispiel HTML, kann weitere Verweise enthalten. Diese können sowohl auf den Archiv-Webserver, als auch ins „Live Web“ zeigen: Insbesondere wenn sich diese in obfuskierter Form (z. B. in Scripten) im Quelltext befinden. Der Betreiber weist explizit darauf hin[17], dass Inhalte ggf. nicht verfügbar sind und aus diesem Grund ein Zugriff auf das „Live Web“ möglich ist. Eine solche nicht angepasste Referenz ist in Abbildung 4.4 bei der Ressource *Rj* zu sehen; die Abbildung wird zu einem späteren Zeitpunkt noch genauer betrachtet und erweitert.

Abrufszenarios

Probleme, die sich aus dem Verweis ins Live Web ergeben, werden in Abschnitt 4.4.3 behandelt. Gleiches gilt für Verweise in die Zukunft, wie sie in Abschnitt 4.4.3 näher beschrieben werden.

Protokollkonservierung

Zusätzliche Header

Neben dem Quelltext und den Ressourcen der Webseite werden auch Pro-

16 Auflistung nur beispielhaft ohne Anspruch auf Vollständigkeit.

17 Archive.org: How did I end up on the live version of a site? or I clicked on X date, but now I am on Y date, how is that possible? https://archive.org/about/faqs.php#202, abgerufen am 14.10.2016.

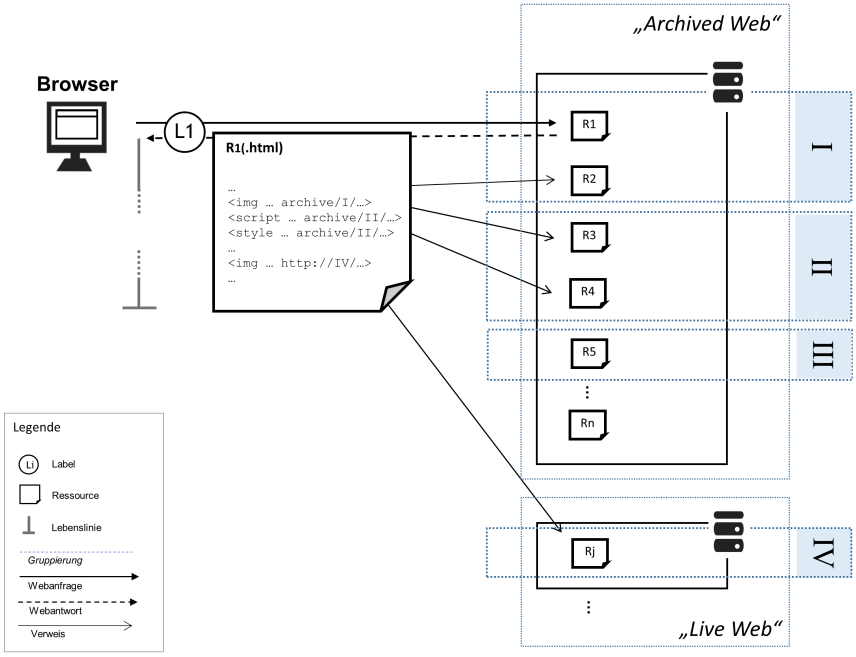

Abbildung 4.4: Browseranfrage einer Webseite (hier `R1.html`) an den Archivserver.

tokolldaten bzw. Headerdaten der ursprünglichen HTTP-Responses gespeichert. Innerhalb der Antwort des Webservers vom Internet Archive sind deshalb zusätzliche HTTP-Felder hinzugefügt worden, dessen Feldnamen mit `X-Archive-Orig-*` beginnen. Beispiele sind:

- X-Archive-Orig-content-type,
- X-Archive-Orig-set-cookie,
- X-Archive-Orig-server,
- X-Archive-Orig-last-modified,
- X-Archive-Orig-expires,
- X-Archive-Orig-connection,
- X-Archive-Orig-etag,
- X-Archive-Guessed-Charset.

Archivierte Protokolldaten

Es ist zu berücksichtigten, dass es sich um die IST-Situation der Analyse aus dem jeweiligen Jahr handelt. Diese kann nur eingeschränkt berücksichtigt werden, wie in Abschnitt 4.4.3 genauer betrachtet wird.

Memento

Memento

Durch Memento RFC 7089 [42] „HTTP Framework for Time-Based Access to Resource States“ wurde im Jahr 2013 ein Protokoll spezifiziert, welches den Umgang mit Archivdiensten durch Einführung eines einheitlichen Protokolls erleichtert. So kann der HTTP-Header auf Client- und Serverseite durch zusätzliche Angaben erweitert werden:

- Clientseitig ist die Angabe eines *Accept-Datetime* im Request-Header möglich, welches den Archivdienst anweist, nur von diesem konkreten Daten Ressourcen zu liefern.
- Der Server informiert den User-Agent mittels *Memento-Datetime* im Response-Header über das Aufnahmedatum einer Ressource.

Memento für retrospektives Browsing

Prinzipiell könnte ein solches Protokoll für eine retrospektive Analyse eingesetzt werden. Weil dazu keine verlässlichen Studien existieren und das dafür notwendige Protokoll mit hohem Aufwand evaluiert werden müsste, ist das Konzipieren einer neuen, verlässlichen Lösung zielführender.

4.4.3 *Einschränkungen bei archivierten Webseiten*

Live-Web

Bei Analysen des „Live Web“ bestehen alle denkbaren Freiheiten in der Vorgehensweise oder Wahl des Forschungsgegenstandes. Bei einer Analyse des *Archived Web* kann nur auf eine *flache Kopie* der archivierten Seiten zurückgegriffen werden. Einschränkungen in der Analyse von Web-Tracking auf archivierten Webseiten werden im Folgenden näher betrachtet.

Fehlende Dialogfähigkeit

Dialoge

Eine Analyse des Dialogs zwischen clientseitig ausgeführten Scripten (JavaScript) und dynamisch generierten Webseiten (z. B. Schnittstellen) ist nicht möglich, da die serverseitige Komponente nicht existiert und aus technischen Gründen nie archiviert werden kann.

Trackingverfahren

Web-Tracking, wie es bereits in Abschnitt 3.3 beschrieben wurde, kann auf unterschiedliche Weise realisiert werden. Bei „Supercookie“-Verfahren, wie sie von Mayer und Mitchell [115] genannt werden, wird eine möglichst persistente Speicherung auf dem System bewirkt, um einen Benutzer über eine größere Zeitspanne wiederzuerkennen. Dabei werden stets Verfahren verwendet, die auch in legitimen Fällen Anwendung finden bzw. rein funktionalen Charakter besitzen. Ein Beispiel hierfür ist der Einsatz von Entitiy Tags:

BEISPIEL: UNTERSUCHUNG VON ETAGS. Entity Tags können nach RFC 2616 [121] für eine Versionierung von Ressourcen (z. B. Bilder) verwendet und damit wiederholtes Nachladen des selben Inhalts verhindert werden. Das *ETag*-Feld des HTTP-Headers kann, wie von Ayenson et al. [9] beschrieben und live demonstriert [110], als Cookie dienen und zum Tracking eingesetzt werden. Ob das Feld tatsächlich zum Tracking genutzt wird, kann nur durch eine Analyse des „Dialogs“ zwischen User-Agent und Webserver festgestellt werden. Es wird geprüft, ob der Inhalt des Feldes sich bei häufigeren (aufeinanderfolgenden) Abrufen unterscheidet. Werden jedem Besucher während einer gewissen Zeitspanne die gleichen E-Tag Informationen zugewiesen, ist eine nachträgliche Unterscheidung und somit der Einsatz zum Tracking

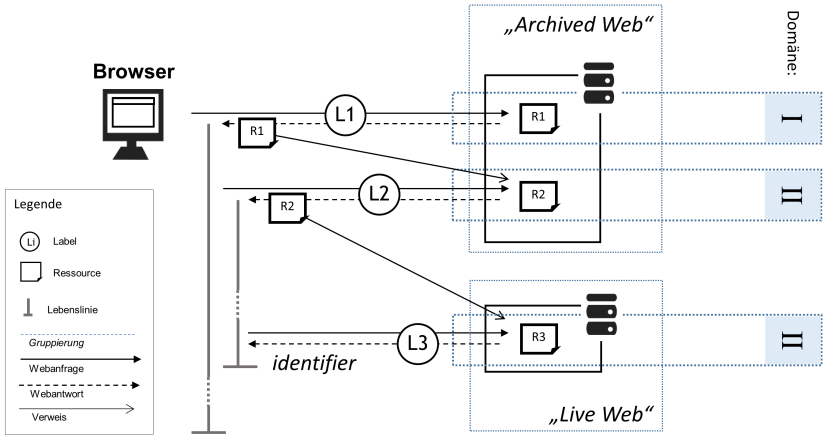

Abbildung 4.5: Generierung einer dynamischen Abfrage (L3) ins Live Web durch Verarbeitung einer Ressource (R2) aus einer Archivdatenbank.

nicht möglich. Variiert das Feld zwischen den Aufrufen mit identischem Inhalt, ist der Einsatz als „Supercookie“ möglich. Allein die Nutzung des HTTP-Feldes, obgleich von der Erst- oder Drittpartei, ist kein Indikator.

Weitere Speicherarten

Dieses Problem setzt sich bei anderen Speicherverfahren fort; so auch beim Last-Modified-Feld oder bei Einsatz von Speicherverfahren mit HTML5. Die reine Nutzung von Speichermethoden durch eine Drittpartei genügt nicht zur Feststellung von Web-Tracking.

Web-Tracking durch mehrteilige Dialoge

Bei Fingerprinting-Verfahren sind keine Speicherungen (z. B. in Form von Cookies) auf dem System notwendig. Diese Verfahren sind in ihrer passiven Form im „Live Web“ nicht stets identifizierbar (Yen et al. [205]). Wird das Verfahren durch clientseitig ausgeführte Skripte unterstützt (vgl. Mulazzani et al. [160]), ist dies eine auffindbare aktive Komponente. Von Nikiforakis et al. [133] werden drei typische Fingerprinting-Anbieter (*BlueCava*, *Iovation ReputationManager*, *ThreatMetrix*) näher betrachtet und ihr Einsatz im „Live Web“ gemessen: Eine solche Vorgehensweise kann auch bei archivierten Webseiten reproduziert werden. Zu berücksichtigten ist, dass solche Bibliotheken durch Obfuskierung versteckt oder umfangreich modifiziert werden können, wodurch eine Analyse erschwert oder verhindert wird.

Verweise ins Live Web

Probleme

Probleme entstehen bei Inhalten, die aufgrund fehlender Berücksichtigung während des Archivierungsvorgangs ins „Live Web“ zeigen wie in Abschnitt 4.4.2 bereits beschrieben wurde. Dies geschieht, wenn Abfragen während der Laufzeit dynamisch durch Skripte erzeugt werden.

Nicht archivierte Inhalte

In Abbildung 4.5 wird dieses Problem dargestellt. Der Abruf einer Ressource *R1* von der Domain *I*, z. B. der Startseite, führt zu einem Nachladen der Ressource *R2*. Diese wurde ebenfalls archiviert und gehört der Domain *II* an.

Dabei kann es sich um ein JavaScript-basiertes Skript handeln. Während der Ausführung dieses Skriptes wird dynamisch eine Verbindung zur Domains *II* erzeugt. Da dieser dynamische Aufruf während der Konservierung durch den Archivdienst nicht berücksichtigt werden konnte, würde der Browser während der Verarbeitung eine Verbindung ins „Live Web“ aufbauen (L_3). Sofern die Analyse der Webseite nur auf ein konkretes Jahr beschränkt wäre, würde eine Kontamination der Ergebnisse stattfinden.

Verweise in die Zukunft

Ebenfalls ist es möglich, dass Verweise in die Zukunft zeigen. Der Archivbetreiber beschreibt diesen Sachverhalt in einer FAQ wie folgt:

> „Not every date for every site archived is 100% complete. When you are surfing an incomplete archived site the Wayback Machine will grab the closest available date to the one you are in for the links that are missing. In the event that we do not have the link archived at all, the Wayback Machine will look for the link on the live web and grab it if available. [...] “
> Quelle: [84].

Fehlverweise

Für Ressourcen, die während des Archivierungsvorgangs nicht berücksichtigt wurden, kann keine nachträgliche Erfassung erfolgen. Stattdessen werden auf Kopien referenziert, die zu einem anderen Zeitpunkt gesichert wurden. Diese können in der Vergangenheit als auch in der Zukunft erstellt worden sein. Als Beispiel kann die Webseite msn.com vom 29. Juni 2006 betrachtet werden:

```
1 http://web.archive.org/web/20060629212141/http://www.msn.com/
```

Wird diese Webseite geöffnet, findet sich u. a. der folgende Verweis:

```
1 http://web.archive.org/web/20110401201722im_/http://c.msn.com/c.
      gif?di=340&pi=7317&ps=83527&tp=http://www.msn.com/&rf=http://
      www.indianainnovation.com/democonfig5.js&MUID=2E08C1B840A76D8
      F0028C07F44A76D23
```

Verweis in die Zukunft

Wie am Zeitstempel innerhalb der Adresse zu erkennen (20110401201722 entspricht 2011-04-01 20:17:22) ist, wird auf das Jahr 2011 verwiesen, obwohl ursprünglich eine Version aus dem Jahr 2006 angefordert wurde.

Folgt man dem Aufruf der Webseite, findet eine Umleitung zu einem (zu dieser Zeit) bekannten Tracker statt:

```
1 http://web.archive.org/web/20110401201722im_/http://c.atdmt.com/c
      .gif?di=340&pi=7317&ps=83527&tp=http://www.msn.com/&rf=http
      ://www.indianainnovation.com/democonfig5.js&RedC=c.msn.com&
      MXFR=3676CF11DEF747F78590898B2C304E7E
```

Sind Tracker innerhalb eingebetteter Skripte versteckt, wurden sie ggf. nicht während der initialen Analyse verarbeitet. Nachträgliche Verarbeitungen können möglicherweise nicht berücksichtigt werden, sofern die Referenz in die Zukunft verweist.

Ausschluss von Fehlverweisen

Es ist eine grundsätzliche Designfrage der Analyse, wie mit diesen Referenzen umzugehen ist. Einerseits bedeutet die Nichtverfolgung, dass Verweise unberücksichtigt bleiben. Werden diese andererseits verfolgt, können sich nicht nachvollziehbare Ergebnisse ergeben wie z. B. Einbettungen von Drittparteien, die im betrachteten Jahr noch nicht existierten.

Um ein klares Bild des jeweiligen Jahres zu erhalten, muss der Verweis in die Zukunft wie ein Verweis ins Live Web bewertet werden.

Unterscheidung Erst- und Drittpartei

Der Begriff Drittpartei wurde in Abschnitt 3.2 erläutert. Im Folgenden wird gezeigt, dass es keine harten technischen Indikatoren gibt, welche die Elemente einer Webseite zweifelsfrei als „intern“ (Erstpartei) oder „extern“ (Drittpartei) kennzeichnen. Am Beispiel *Facebook* wird gezeigt, wie schwierig eine solche Unterscheidung im „Live Web“ ist.

Identifizierung von Betreibern

Während aus Benutzersicht hinter der Web-Adresse facebook.com ein Unternehmen steht, ist dieses soziale Netzwerk auch unter anderen Adressen erreichbar. Eine Änderung der Top-Level-Domain, z. B. *facebook.net* anstelle *facebook.com*, wird häufig als der gleiche Ansprechpartner wahrgenommen. Neben der Änderung der Top-Level-Domain gibt es allerdings auch etwas weniger offensichtliche Beispiele wie *fbcdn.net*: Während im Falle von Facebook i.d.R. *Facebook.com* (oder .net) vom Benutzer abgerufen wird, befinden sich die Ressourcen überwiegend im *Content Delivery Network* (CDN) von Facebook. Der Name der Domains suggeriert, dass dieses Netzwerk speziell für Facebook-Inhalte vorgesehen ist und legt daher nahe, dies ebenfalls als zu Facebook angehörende interne Ressource zu sehen. Grundsätzlich würde an dieser Stelle eine Third-Level-Domain (z. B. cdn.facebook.com) die gleichen technischen Möglichkeiten bereitstellen. Für eine solche Auslagerung lassen sich technische und organisatorische Gründe leicht finden.

Registrierungsdaten

Einen Hinweis, ob eine Domain zu einem Unternehmen gehört, gibt die jeweilige Registrierungsstelle. Die *Internet Corporation for Assigned Names and Numbers* (ICANN) legt für jede Top-Level-Domain die zuständige Registrierungsstelle fest. So ist für alle .de-Adressen die *DeNIC* in Frankfurt zuständig; für .com, .net und weitere die InterNIC. Diese Registrierungsstellen lassen Abfragen über die hinterlegten Stammdaten für technische und administrative Zwecke zu. Am Beispiel Facebook lässt sich zeigen, dass sich diese Daten nur begrenzt zur genaueren Identifikation und Gruppierung verwenden lassen. So ist für *facebook.com* und *facebook.net* das Unternehmen (Stand: 2016) „MARKMONITOR INC.“ hinterlegt: ein Unternehmen, welches sich seit 1999 dem Schutz von Markennamen im Internet verschrieben hat. Auf diesem Weg soll präventiv Typosquatting vermieden werden: die Registrierung von Tippfehlerdomains, um auf dieser Werbung oder möglicherweise schadhaften Inhalt zu platzieren. Auch wenn die Registrierung der Domain durch einen solchen Dienstleister eher die Ausnahme bleibt, zeigt es die nur eingeschränkt sinnvolle Verwendbarkeit der hinterlegten Stammdaten auf.

IP- und DNS-basierte Identifikation

Eine alternative Möglichkeit bietet der genutzte IP-Adressbereich oder der

zuständige DNS-Server wie es in den Arbeiten von Libert [107] sowie Englehardt und Narayanan [48] der Fall ist. Insbesondere bei größeren Anbietern liegt es nahe, dass alle Domains den gleichen Nameserver verwenden. So verwenden die Facebook-Domains *facebook.com*, *facebook.net*, *fbcdn.net* und auch die Tippfehlerdomains *faebook.com* und *facbook.com* den DNS-Server *A.NS.FACEBOOK.COM*. Bei diesem Vorgehen besteht die Gefahr, kleinere Anbieter fälschlicherweise zu einer großen Gruppe zusammenzufassen. Ein Beispiel ist die Strato AG mit Sitz in Berlin, die nach eigener Aussage 4 Million Domains verwaltet[18] und u. a. E-Mail- und Hosting-Pakete anbietet. Dabei wird, sofern vom Benutzer oder durch das Paket nicht anders definiert, der Nameserver NS.STRATOSERVER.NET verwendet. Die gleiche Problematik zeigt sich bei der Verwendung von IP-Adressinformationen, die bei der ICANN bzw. dem untergeordneten RIPE Network Coordination Centre (RIPE CNN) erfragt werden können.

Impressum

Da weder IP-Adresse noch der DNS-Eintrag bzw. die Domain eindeutige Informationen liefert, verbleibt nur die Anwendungsebene: ein Impressum (oder sonstige Kontaktinformationen) auf der Webseite kann weitere Auskunft über die verantwortliche Stelle liefern. Im Folgenden werden interne Ressourcen als all jene gesehen, die zum Unternehmen bzw. zur Person gehören, die für den Webauftritt verantwortlich ist – auch wenn dies nicht in allen Fällen durch technische Verfahren in Erfahrung gebracht werden kann.

4.5 Anforderungen an die technische Implementierung

Anforderungen

Durch die genaue Analyse der zur Verfügung stehenden Archivdaten in Abschnitt 4.4 konnten Anforderungen an die technische Umsetzung identifiziert werden. Diese müssen bei der Auswahl und Implementierung des Analysewerkzeugs in Kapitel 5 berücksichtigt werden:

(A-1) **Vollständigkeit**. In Abschnitt 4.4.3 wurden Einschränkungen bei Untersuchungen des „Archived Web“ näher betrachtet. So sollten Auswertungen vermieden werden, die nur auf einen Teil der Testmenge verlässlich anwendbar sind, oder wenn diese nur in bestimmten Fällen Resultate erzielen. Aus diesem Grund empfiehlt es sich, nur Analysen vorzunehmen, die ein akzeptables Maß an Vollständigkeit gewährleisten.

(A-2) **Kontaminationsfreiheit**. Die Verwendung von Archivmaterial ermöglicht die Wahl eines speziellen Datums oder einer Zeitspanne. Sofern Webseiten eines bestimmtes Tages, Monats oder Jahres betrachtet werden, dürfen diese nicht mit Inhalten aus anderen Jahrgängen oder sogar aus dem „Live Web“ kontaminiert werden. Es muss sichergestellt sein, dass die erhobenen Daten allein aus der gewählten Zeitperiode stammen.

(A-3) **Effizienz**. Ziel ist es, eine mit anderen Publikationen auf diesem Gebiet (Abschnitt 3.3) vergleichbare Erhebung zu erzielen.

18 https://www.strato.de/press/#daten_u_fakten, abgerufen am 07.11.2016.

Dabei muss die technische Implementierung so gewählt werden, dass keine vermeidbaren Verzögerungen während der Ausführung entstehen. Es muss berücksichtigt werden, dass der Erfassungsaufwand 10 bis 16 Mal höher ist als bei anderen quantitativen Studien, da im Idealfall die Webseite für alle verfügbaren Jahrgänge geladen werden muss. Aus diesem Grund muss die technische Umsetzung ein effizientes Erfassen und Verarbeiten ermöglichen.

(A-4) **Auswertbarkeit**. Die technische Umsetzung muss eine leichte und nachvollziehbare Auswertbarkeit der erhobenen Daten ermöglichen. Eine Orientierung an gängigen Standards („Best practice“) ist empfehlenswert.

5 ENTWURF UND IMPLEMENTIERUNG DES WERKZEUGS

Zusammenfassung: Nach einer Betrachtung bestehender Werkzeuge und Werkzeugklassen aus verwandten Arbeiten findet die Auswahl eines Messwerkzeugs statt. Dieses wird unter Berücksichtigung der Anforderungen aus Kapitel 4 analysiert, modifiziert und erweitert. Abschließend werden die gestellten Anforderungen auf Umsetzung überprüft.

5.1 EINLEITUNG ZUR WERKZEUGENTWICKLUNG

Kapitelübersicht

Nachdem in Kapitel 4 das Datenmaterial sondiert und Randbedingungen einer retrospektiven Studie identifiziert wurden, wird im Folgenden die Implementierung eines dafür notwendigen Werkzeugs durchgeführt. Zu diesem Zweck wird in Abschnitt 5.2 die technische Ausgangslage betrachtet, um anschießend ein Werkzeug auszuwählen, das den Anforderungen aus Abschnitt 4.5 gerecht wird und entsprechend erweitert werden kann. Die Implementierung des Messwerkzeugs wird in Abschnitt 5.4 beschrieben. Abschließend wird die Umsetzung der Anforderungen in Abschnitt 5.5 geprüft.

5.2 ANALYSE DER TECHNISCHEN AUSGANGSLAGE

Übersicht

Die Ausganglage wird durch eine Erfassung der Werkzeuge in verwandten Arbeiten analysiert. Diese können anschließend in Äquivalenzklassen eingeteilt werden.

5.2.1 *Analysewerkzeuge in verwandten Arbeiten*

Aktuelle Werkzeugformen

Innerhalb der verwandten Arbeiten (Abschnitt 3.3) werden verschiedene Techniken zur Erfassung und Analyse von Internetseiten in Bezug auf Web-Tracking eingesetzt. Dabei sind vor allem die Veröffentlichungen von Interesse, die eine größere Anzahl ($\geq$ 10 000) von Webseiten erfassen. Anschließend werden Äquivalenzklassen von Techniken vorgestellt, die aus diesen Publikationen gebildet werden können. Diese Klassen sind nicht disjunkt, weil sie in vielen Fällen aus mehrere Techniken zusammensetzen.

Analyse

Im Folgenden werden die Werkzeuge aus bestehenden Arbeiten analysiert und gruppiert. In Abschnitt 5.2.2 findet eine Klassifikation der Werkzeuge und eine Betrachtung der Vor- und Nachteile statt.

BROWSERAUTOMATISIERUNG. Ein Browser ist üblicherweise auf Interaktion mit dem Benutzer über die grafische Benutzeroberfläche ausgelegt. Um solche Automatisierungen ohne Benutzeraktionen (maschi-

nell) zu ermöglichen, kann das Automatisierungsframework Selenium [162] eingesetzt werden. Dieses ermöglicht die Ansteuerung diverser Browser: (Google) Chrome bzw. Chromium, (Mozilla) Firefox, PhantomJS, Safari und weitere. Acar et al. [1] verwenden u. a. Selenium mit Chromium zur Analyse von Fingerprintingverfahren auf Webseiten; Englehardt und Narayanan [48] nutzen dieses ebenfalls in Verbindung mit dem Mozilla Firefox Browser.

SPEZIELLER BROWSER. Libert [107] verwendet den Browser *PhantomJS* [75]. Dabei handelt es sich um einen vollständigen (s.g. „full web stack“) Browser, der alle üblichen Prozesse innerhalb eines Browsers nachbildet (z. B. die Interpretation und Ausführung der JavaScript-Dateien), jedoch keine grafische Ausgabe vorsieht und Schnittstellen für eine maschinelle Auswertung liefert.

BROWSERMODIFIKATION. Neben dem Chrome Browser steht auch eine freie Open Source Variante namens *Chromium* [82] zur Verfügung. Diese lässt sich für eigene Zwecke anpassen und kompilieren. Von Stopczynski et al. [175] wird diese Möglichkeit zur Entwicklung einer Schutzmaßnahme mit anschließender Evaluation genutzt.

BROWSERERWEITERUNG. Ein Browser kann durch ein Addon (z. B. für den Mozilla Firefox oder Google Chrome) erweitert werden. Ist eine solche Erweiterung implementiert und installiert, ist nur das Öffnen der zu analysierenden Webseite notwendig. Von Roesner et al. [153] wurde der Chrome-Browser um das Addon „TrackingObserver“ [185] erweitert. Ähnlich sind Acar et al. [2] und Mayer und Mitchell. [115] vorgegangen.

Eine Automatisierung (z. B. über Selenium) ist zwar möglich, aber nicht zwangsläufig notwendig. So wird bei Falahrastegar et al. [52] der Chrome Browser mit einer entsprechenden Erweiterung über ein eigenes python-Script automatisiert.

PROXY. Die Verwendung eines Proxys in Verbindung mit einem gängigen Browser ist ebenfalls möglich. Krishnamurthy et al. [96] haben für ihre Analyse anfänglich ein Proxy verwendet, der erst später zu einer Erweiterung für den Mozilla Firefox wurde. Diese Netzwerk-basierte Erfassung kam auch bei Abdelberi et al. [30] zum Einsatz.

Ist eine detailliertere Auswertung der Netzwerkdaten notwendig, wird der BrowserMob[1] Proxy bei der Verwendung des Automatisierungsframeworks Selenium eingesetzt. Seit der Version 3 von Selenium ist dies nicht länger notwendig.

EXTERNE ANWENDUNGEN. In einigen Fällen liegen die Daten außerhalb des Erfassungsraums des Browser wie bei der Erweiterung Adobe Flash. In diesem Fall muss eine separate Anwendung zur Analyse entwickelt werden, wie bei McDonald und Cranor [117] beschrieben.

EIGENE CRAWLER. Reichen die Rohdaten der Webseiten für einen Analyseprozess aus und sind die Verarbeitungsfähigkeiten eines Browsers

1 BrowserMobProxy, https://bmp.lightbody.net/, abgerufen am 16.11.2017.

nicht notwendig, können über eingebaute Bibliotheken in Programmiersprachen die Daten abgerufen werden. Dabei werden die Quelltexte von Webauftritten erfasst wie beispielsweise von Schelter und Kunegis [159] durchgeführt.

5.2.2 *Klassifikation bestehender Werkzeuge*

Äquivalenzklassen

Die Betrachtung der bestehenden Arbeiten zeigt, dass die eingesetzten Techniken sehr vielfältig sind. Eine dominierende Analyseform wurde nicht festgestellt. Allerdings ist auf dieser Basis eine Äquivalenzklassenbildung möglich, welche die verwendeten Techniken einteilt.

STATISCHE ANALYSE. Bei dieser Analyseklasse findet nur eine textuelle Analyse der Rohdaten statt. Die verfügbaren Quelltexte werden mittels regulärer Ausdrücke oder String-Matching-Algorithmen (z. B. Fuzzy-Suche) analysiert und bewertet. Basis dieser Erhebung sind Abfragebibliotheken, die von Programmiersprachen bereitgestellt werden (z. B. `urllib` in Python), oder so genannte „Web Crawler“, welche das automatisierte Herunterladen und die Verwaltung der geladenen Webseiten sowie deren Verarbeitung ermöglichen. Die Vorteile dieser Verarbeitungsform ist die hohe Effizienz und Sicherheit der Untersuchung. Werkzeuge zu statischen Analyse sind:

- Scrapy[2]: ein Framework zum Extrahieren von Daten aus Webseiten.
- Apache Nutch[3]: ein Werkzeug, welches für das Common Crawl Projekt verwendet wird, das bereits in Abschnitt 4.3.1 als Datensammlung erwähnt wurde.
- Heritrix[4]: wird in weiten Teilen für die Datensammlung von archive.org eingesetzt.

DYNAMISCHE ANALYSE. Bei dieser Form der Analyse findet neben dem Beziehen der Komponenten einer Webseite auch eine Interpretation und Ausführung von Inhalten statt. Diese werden innerhalb von Browsern und in der Literatur als „Aktive Inhalte“ bezeichnet. Dazu gehört die prioritäre Software ActiveX (von Microsoft), Flash (Macromedia, später Adobe), Java (Sun, später Oracle) sowie das aktuell vorherrschende JavaScript, dessen standardisierte Form als ECMAScript bekannt ist. Diese aktiven Inhalte ermöglichen eine Veränderung der Webseite, während und nach dem Ladevorgang. Sie können darüber hinaus weitere Verbindungen zu Drittparteien bewirken, die bei einer reinen Analyse der Rohdaten unberücksichtigt bleiben würden. Nachteilig ist, dass die Verarbeitung mehr Zeit in Anspruch nimmt und sich

2 Scrapy | A Fast and Powerful Scraping and Web Crawling Framework https://scrapy.org/, abgerufen am 14.01.2017.
3 Apache Nutch http://nutch.apache.org/, abgerufen am 14.01.2017.
4 Heritrix - http://crawler.archive.org/index.html, abgerufen am 14.01.2017.

die Ausführung von Programmen und Skripten aus Sicherheitsgründen als problematisch herausstellen kann.

Analysen, die solche aktiven Inhalte einschließen, lassen sich in zwei Unterklassen unterteilen.

HEADLESS BROWSER. Bei diesen Analysewerkzeugen werden die Verhaltensweisen eines handelsüblichen Browsers weitgehend nachgebildet, sie sind allerdings durch Programmtext steuer- und konfigurierbar. Anders als normale Browser, sind sie deutlich leichtgewichtiger aufgebaut und bestehen überwiegend aus dem Rendering-Modul (vgl. Abschnitt 2.6) selbst. Vorteile bei der Nutzung solcher Bibliotheken ist, dass die Automatisier- und Auswertbarkeit bereits bei der Entwicklung vorgesehen wurde. Nachteilig ist, dass je nach Analyse eine gewisse Unklarheit der vollen Kompatibilität herrscht. Beispiele für Werkzeuge dieser Kategorie sind:

- Der „Full web stack“ Browser PhantomJS[5]. Ein über JavaScript ansteuerbarer Kommandozeilenbrowser.
- Das Qt WebKit[6]. Bei Qt handelt es sich um eine plattformübergreifende Programmbibliothek, die in verschiedenen Programmiersprachen genutzt werden kann. Die darin enthaltene WebKit-Bibliothek ermöglicht das Laden und Interpretieren von Webseiten und beherrscht die üblichen Internetstandards.
- Die Programmbibliothek HtmlUnit[7] ist ein in Java programmierter Browser, der als Modul in eigenen Applikationen eingesetzt werden kann.

STANDARD BROWSER. Um die Analyse möglichst realitätsgetreu zu gestalten, liegt den Einsatz der technischen Mittel nahe, die auch ein üblicher Internetnutzer verwenden würde. Nachteilig ist, dass der Browser aus vielen Programmkomponenten besteht, wodurch die Ausführung vergleichsweise langsamer und der Ressourcenverbrauch deutlich höher ist.

In der Vergangenheit (Stand: 2017) mussten Automatisierungen durch Browsererweiterungen oder durch spezielle Anpassungen am Browser umgesetzt werden. Diesen Mangel haben die Hersteller erkannt und erleichtern die Steuerung durch Programmcode: Das Marionette[8] Protokoll (Firefox 45.5.1, veröffentlicht September 2016) ermöglicht eine externe Steuerung und Automatisierung des Firefox Browsers.

5 PhantomJS http://phantomjs.org/, abgerufen am 14.01.2017.
6 QtWebKit Guide http://doc.qt.io/qt-4.8/qtwebkit-guide.html, abgerufen am 14.01.2017.
7 http://htmlunit.sourceforge.net/, abgerufen am 14.01.2017.
8 https://developer.mozilla.org/en-US/docs/Mozilla/QA/Marionette, abgerufen am 30.11.2017.

5.2.3 *Erkennung von Web-Tracking*

Erkennung im Archiv

Es ist offensichtlich, dass nur eine Auswertung der Daten möglich ist, die vom Archivierungsdienst zum Zeitpunkt der Speicherung gesichert wurden. Wie mit Abbildung 4.5 verdeutlicht wurde, können sich durch dynamisch nachgeladenen Inhalte Aufrufe ins „Live Web“ ergeben. Diese sind in der Lage weitere Ressourcen von weiteren Drittparteien nachzuladen. Darüber hinaus wurden in Abschnitt 4.4.3 Einschränkungen gezeigt, welche eine klare Identifikation von Web-Tracking-Verfahren auf archivierten Webseiten erschwert.

Erkennung über Drittparteien

In aktuelleren Publikationen wird die Einbettung einer Drittpartei und die damit immer verbundene Weitergabe der IP-Adresse des Benutzers als hinreichendes Indiz für Web-Tracking (oder dessen Möglichkeit) angenommen wie bei Libert [107]. Während auf diese Weise eine genauere Differenzierung der Verfahren unterbleibt, werden alle Informationen zu eingebetteten Dritten während des Aufrufs der Webseite gesammelt. Dies insbesondere auch um dem Anspruch auf Vollständigkeit (A-1) gerecht zu werden.

Weiergabe von IP Daten

Es existieren keine „harten Indikatoren“, um Hosts einer Domains zuzuordnen (vg. Intern vs. Extern 4.4.3). Eine Auswertung von DNS- oder Registrierungsinformationen schließt sich bei der Durchführung einer retrospektiven Analyse aus, da diese nicht für eine größere Zeitspanne rückwirkend einsehbar sind. Dienste, die eine Aufzeichnung dieser Daten bieten (vgl. DNSTrails[9]), haben ihre Aufzeichnung vergleichsweise spät begonnen oder sich als unzuverlässig erwiesen. Eine automatische Suche nach Impressumsinformationen ist nicht gewinnbringend, da diese nicht von allen Ländern gefordert werden.

Unterscheidung über Domain

Infolgedessen können eingebettete Hosts einer Webseite nur mit der Second Level Domain zwischen intern und extern unterschieden werden. Es ist nicht zu erwarten, dass eine Domains *tld1.A.net* eine Einbettung durchführt, die nach *tld2.A.net* zeigt, obwohl sie zwei verschiedenen Unternehmen angehört. Des Weiteren ist nicht davon auszugehen, dass zwischen der Einbettung auf *A.net* zu *A.com* ein Unterschied der Zugehörigkeit besteht, sofern *A.net* eine Einbettung zu *A.com* unternimmt. Bleibt also die Second Level Domain identisch (hier: A), handelt es sich um das gleiche Unternehmen und deshalb um eine interne Ressource. Findet zwischen *A.com* und *A.net* keine Kommunikation in Form einer Einbettung statt, werden diese als verschieden betrachtet.

5.3 Entwurf eines Analysewerkzeugs

Anforderungen

In Abschnitt 4.5 wurden abstrakte technische Anforderungen an eine mögliche Implementierung gestellt. Diese müssen für die Entwicklung einer neuen oder zur Erweiterung einer bestehenden Lösung berücksichtigt werden. In Hinblick auf die in Abschnitt 5.2.1 vorgestellten Werkzeuge stellt sich

9 http://dnstrails.com/, abgerufen am 15.03.2018.

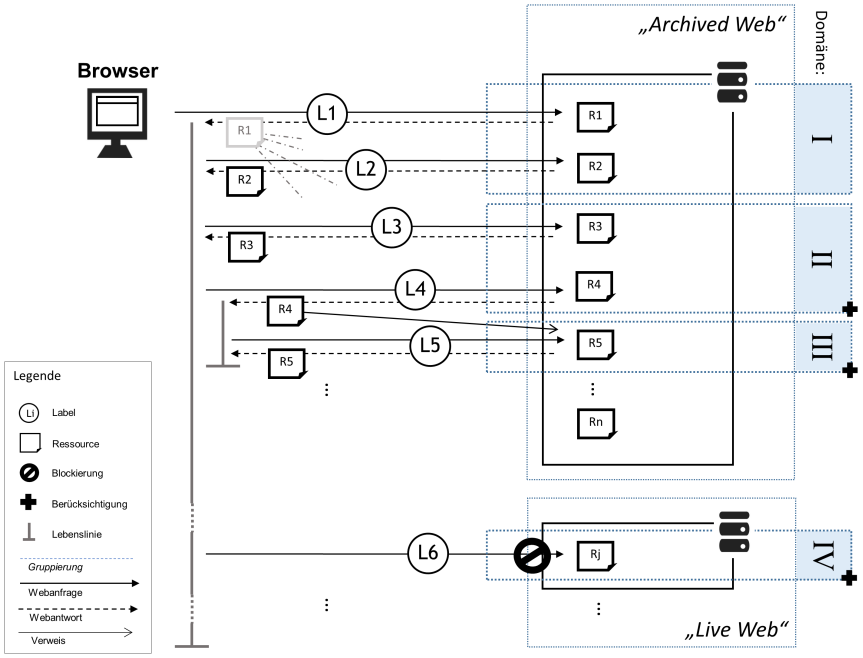

Abbildung 5.1: Browseranfrage einer Webseite (hier `R1.html`) an den Archivserver.

heraus, dass diese nicht direkt auf archivierte Webseiten eingesetzt werden können und einer Anpassung bedürfen.

Prüfung des Werkzeugtyps

Um die Kontaminiationsfreiheit (A-2) zu gewährleisten, muss der Browser daran gehindert werden, Ressourcen von beliebigen Stellen nachzuladen. An einer möglichst zentralen Stelle müssen alle Anfragen bewertet und ggf. blockiert werden, wenn diese die Analyse beeinflussen könnten. Abbildung 5.1 zeigt, wie eine Verbindung (L6) ins „Live Web“ blockiert wird.

5.3.1 *Auswahl der Werkzeugklasse*

Kriterien der Auswahl

In Abschnitt 5.2.2 wurden verschiedene Werkzeugklassen beschrieben. Für die retrospektive Analyse ist eine reine statische Analyse der Quelltexte ungeeignet, da die Vollständigkeit (A-1) einer Analyse nicht gewährleistet wäre.

BEISPIEL: DEAKTIVIERTES JAVASCRIPT AUF HUFFINGTONPOST.DE

In einem Test wurde die Webseite http://huffingtonpost.de daran gehindert, aktive Inhalte (JavaScript) auszuführen, indem JavaScript innerhalb des Browsers deaktiviert wurde. Eine Messung mit der Browsererweiterung Lightbeam [109] (mit deaktivierter Tracking Protection) am 13.01.2017 zeigte Verbindungen zu sechs Anbietern: huffingtonpost.de, scorecardresearch.com, quantserve.com, imrworldwide.com, spotxchange.com spotxcdn.com.

Das Zulassen von JavaScript führt zu Verbindungen zu insgesamt 39 Anbietern: huffingtonpost.de, mediavoice.com, smartredirect.de, googletagservices.com, smartadcheck.de, polarmobile.com, scorecardresearch.com, yieldlab.net, rqtrk.eu, chartbeat.net, criteo.com, xplosion.de, doubleclick.net, huffingtonpost.com, bf-ad.net, fbcdn.net, spotxchange.com, googlesyndication.com, twitter.com, facebook.com, amazon-adsystem.com, imrworldwide.com, quantserve.com, adnxs.com, nr-data.net, babator.com, ioam.de, parsely.com, pinterest.com, google.com, revsci.net, aol.com, huffpost.com, gstatic.com, semasio.net, google-analytics.com, emetriq.de, aolcdn.com, spotxcdn.com.

Schlussfolgerung

Anhand dieses Beispiels werden die Unterschiede der Drittparteieinbettung sichtbar, wenn aktive Inhalte zugelassen oder blockiert werden. Um eine Vergleichbarkeit zu gewährleisten, müssen diese innerhalb der retrospektiven Analyse berücksichtigt werden. Aus diesem Grund wird eine dynamische Analyse durchgeführt.

Werkzeugwahl

Nachdem nun im ersten Schritt sich für eine dynamische Analyse entschieden wurde, wird im nächsten Schritt wird geprüft, ob ein *Headless Browser* verwendet werden kann, oder ob der Einsatz eines *Standard Browser* notwendig ist. Dies ist von der Entscheidung abhängig, welches Ziel die Analyse verfolgt. Dabei kommen zwei Ziele in Betracht:

(ZI-1) Einzelne Webseiten sollen intensiv auf den Einsatz von Web-Tracking (im Sinne von Drittparteieinbettungen) untersucht werden (qualitativ), oder

(ZI-2) es sollen möglichst viele Webseiten betrachtet werden, um ein Gesamtbild zu generieren (quantitativ).

Vollständigkeit vs. Effizienz

So ist eine Abwägung notwendig, die Gewichtung zwischen Vollständigkeit (A-1) und Effizienz (A-3) entsprechend zu verteilen. Bei einer Analyse des „Live Web“ ist meines Erachtens die Genauigkeit (ZI-1) der Geschwindigkeit (ZI-2) vorzuziehen. Bei der retrospektiven Analyse muss hinterfragt werden, welchen Mehrnutzen eine intensive Analyse einer bereits archivierten Webseite hat. In Abschnitt 4.4.3 wurde beschrieben, welchen Einschränkungen die retrospektive Betrachtung von Webseiten unterliegt. Aufgrund der Analyse einer „flachen Kopie“ einer Webseite können viele Web-Tracking-Techniken nicht rückblickend nachvollzogen werden. Die Vollständigkeit ist aus diesem Grund anzuzweifeln, selbst wenn man die intensivere Variante (ZI-1) wählen würde.

Begründung der Wahl

Im vorherigen Abschnitt 5.2.3 wurde Web-Tracking auf die Einbettung von Drittparteien reduziert. Da diese Daten von beiden Techniken zuverlässig erbracht werden kann, wird die effizientere Technik *Headless Browser* eingesetzt.

5.3.2 *Berücksichtigung der Datenquelle*

In Abschnitt 4.4 wurde *archive.org* als Datenquelle gewählt und analysiert. Referenzen werden innerhalb der Webseite auf das Archiv selbst umgebogen, wie in Abschnitt 4.4.2 an einem Beispiel demonstriert wurde.

Verweistypen

Auf diesen archivierten Webseiten zeigen sich sechs verschiedene Typen von Anfragen, die jeweils unterschiedlich behandelt werden müssen. Eine erste Unterteilung zeigt sich bei Aufrufen, ob diese ins Archiv selbst abzielen (Typen I bis IV) oder auf eine Ressource im „Live Web" (Typen V und VI). Zielen die Aufrufe auf ein Ressource im „Live Web", muss diese bekanntermaßen blockiert werden. Es ist unstrittig, dass die Webseite einen Versuch der Einbettung zu dieser Domain unternommen hat und diese nur beim Archivierungsvorgang nicht berücksichtigt wurde. Ist dies der Fall kann dieser Verweis als Einbettung gezählt werden, obwohl das dahinter liegende Element nicht mehr auswertbar ist.

Intern vs. Extern

Das nächste Unterscheidungskriterium is die Einteilung in eine interne oder externe Ressource. Die Problematik dieser Differenzierung wurde in Abschnitt 4.4.3 behandelt.

Zeitpunkt

Das letzte Unterscheidungskriterium zielt auf den Zeitpunkt der Archivierung der Ressource ab. Auch bei einem archivierten Element kann dieses nicht bedenkenlos ausgewertet werden, da Ressourcen in der Zukunft liegen können. Das bedeutet, dass bei Betrachtung einer Webseite aus dem Jahr 2000 eine Verlinkung auf eine Version im Jahr 2013 führen könnte. Dies muss aus offensichtlichen Gründen verhindert werden, weshalb die Zeit in die Ressource einbezogen wird.

Anfragetypen

In Tabelle 5.1 sind die Anfragetypen aufgezeigt, sowie die Entscheidung wann diese blockiert[10] und wann gezählt[11] werden.

Typ	Anfrage	Gezählt	Blockiert
I	A/web/<(*zeit* ≤ *jahr*)>/<int domain>/<res>	Nein	Nein
II	A/web/<(*zeit* > *jahr*)>/<int domain>/<res>	Nein	Ja
III	A/web/<(*zeit* ≤ *jahr*)>/<ext domain>/<res>	Ja	Nein
IV	A/web/<(*zeit* > *jahr*)>/<ext domain>/<res>	Ja	Ja
V	<int domain>/<res>	Nein	Ja
VI	<ext domain>/<res>	Ja	Ja

Tabelle 5.1: Überblick der Anfragetypen – A steht für http(s)://web.archive.org

5.3.3 *Parallelisierung*

Prozessbasierte Verkapselung

Jede Anfrage an das Archivsystem stellt einen in sich geschlossenen Pro-

10 Im Sinne von einem Herunterladen und Auswerten der Ressource.

11 Im Sinne als externe Ressource berücksichtigt.

grammablauf dar, der nach Eingabe einer URI selbstständig ohne weitere Kommunikation agieren kann und abschließend das Ergebnis zurückgibt. Solche Prozesse, die einen vergleichsweise geringen Kommunikationsbedarf mit dem restlichen System haben, eignen sich zur parallelisierten Ausführung. Bei der Analyse einer größeren Anzahl von Webseiten können Programmfehler dazu führen, dass Webseitenabrufe nicht terminieren. Infolgedessen werden die Abfragen nicht nur auf Thread-Ebene verkapselt, sondern in eigenständige Prozesse ausgelagert.

Ablage der Daten

Der Parser ruft als eigener Prozess eine Archivwebseite ab und prüft diese auf externe Einbettungen, wie in Abschnitt 5.3.2 beschrieben wurde. Anschließend wird das Ergebnis separat im Dateisystem abgelegt. Es können Fehlercodes enthalten sein, wenn der Prozess noch in der Lage war, diese zu schreiben. Während der Ausführung wird der Vorgang vollständig überwacht. Wenn die Aufgabe beendet ist, werden die Ergebnisse eingelesen und mit den anderen verknüpft.

Überwachung

Für jeden gestarteten Prozess wird in der Steuerungseinheit ein Timer initiiert, der bei Überschreitung eines Maximalwertes den Prozess terminiert. Im Fehlerfall beendet sich der Prozess selbst, gefolgt von der Rückgabe eines Fehlercodes, oder wird aktiv von der Steuerung beendet.

5.3.4 *Persistente Speicherung*

Anforderungen an die Speicherung

An ein Speicherungssystem des hier entworfenen Werkzeugs werden verschiedene Anforderungen gestellt. So kann es während der Analyse der Webseiten zu Netzwerk- oder Systemabbrüchen kommen. Die gespeicherten Daten müssen in so einem Fall konsistent bleiben. Darüber hinaus wird ebenfalls eine leichte und reproduzierbare Auswertung der Daten möglich sein.

Die folgenden Entitäten und Kardinalitäten müssen berücksichtigt werden und sind in Abbildung 5.2 dargestellt:

- Eine Testmenge ist zum Beginn der Analyse festzulegen. Ziel ist es, eine konkrete Anzahl an Webseiten im Archiv zu analysieren – beispielsweise 1000 Webseiten.
- Für jede Webseite wird ein Verlauf über die Jahre zwischen 2000 und 2015 erstellt. Aus diesem Grund gibt es im Idealfall für jede Webseite 16 verschiedene Versionen im Archivsystem, die sich auf das jeweilige Jahr beziehen: also 1000 * 16.
- Diese 16.000 Webseiten werden vom Analysesystem geprüft. Jede dieser Adressen kann zu einem fehlerhaften Ergebnis führen. Beispielsweise weil die Analyse nicht terminiert oder sich aus unbekanntem Zustand selbst beendet hat.
- Jede Webseite erzeugt Netzwerkanfragen. Diese richten unterschiedlich viele Anfragen an Adressen, die an das Archiv oder ins Live Web gerichtet sind. Diese Anfragen werden innerhalb der Datenbank gesammelt.

Daraus ergeben sich die folgenden Entitäten:

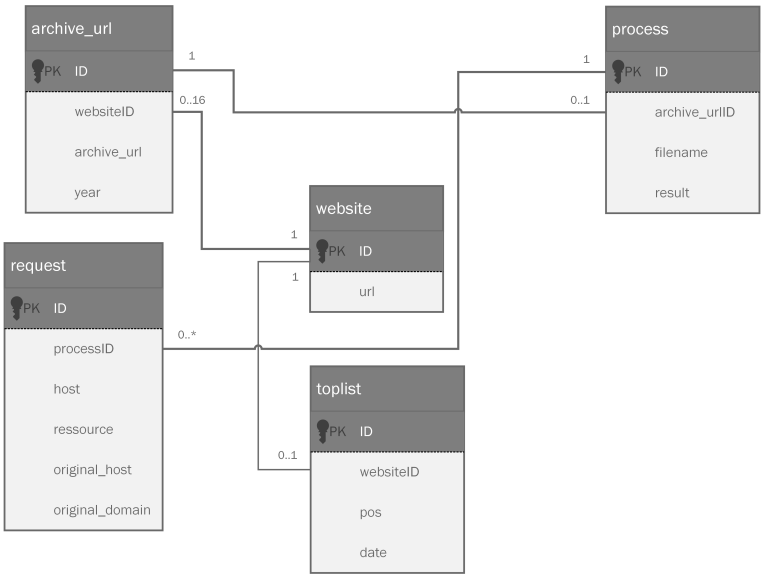

Abbildung 5.2: ER-Diagramm (mit Attributen) gemäß *UML Database Notation*

TOPLIST Eine Sammlung von Webseiten, die als Testmenge in Frage kommen (ggf. sortiert nach Popularität).

ARCHIVE_URL Die Verfügbarkeit von Webseiten auf archive.org und deren Internetadresse pro Jahr.

PROCESS Auflistung der Analysevorgänge und der damit verbundenen Daten (Rückgabe bzw. Ergebniswerte).

REQUEST Externe Verbindungen für jede Analyse (process) und das jeweilige Ziel.

5.4 IMPLEMENTIERUNG

5.4.1 *Auswahl der Messapplikation*

Anforderungen

In Abschnitt 5.3.1 wurde „Headless Browser“ als Werkzeugklasse gewählt. Unter Berücksichtigung der Anforderungen aus Abschnitt 4.5 muss im Folgenden eine konkrete Technik gewählt und für den Zweck der Analyse angepasst werden. Hierbei ist insbesondere die Anforderung der Kontaminationsfreiheit (A-2) zu berücksichtigen. Das gewählte technische Mittel muss in der Lage sein, Verbindungen nur zu bestimmten Webseiten zuzulassen. Verbindungen ins Live Web, wie in Abbildung 5.1 zu sehen ist, müssen blockiert werden. Darüber hinaus muss es möglich sein, alle erstellten Verbindungen, die sich durch das Auswerten einer Webseite ergeben, speichern zu können.

Übersicht

Zunächst wird geprüft, ob und wie die identifizierten „Headless Browser“ angepasst werden können, um die gestellten Anforderungen zu erfüllen:

PHANTOMJS. Die Betrachtung der Schnittstellen des PhantomJS Browsers ermöglichen, alle erstellten Netzwerkverbindungen durch Auswertung des *onResourceRequested* Events zu listen. Gemäß der Spezifi-

kation[12] kann über das Eventobjekt *networkRequest* die Verbindung unterbrochen werden.

QT WEBKIT. Gemäß der Dokumentation[13] werden alle Netzwerkverbindungen über die *createRequest*-Methode der *QNetworkAccessManager*-Klasse geleitet. Diese kann durch eine eigene abgeleitete Klasse überschrieben werden.

HTMLUNIT. Für HtmlUnit konnte keine Lösung ermittelt werden, ohne eine umfangreichere Überarbeitung des Quellcodes vorzunehmen.

Weitere

Weitere Anwendungen wie *Zombie.js* für *Node.js* und *CasperJS* zeigten sich als weniger verbreitet. Den python-basierten Webcrawler *Scrapy* kann mittels *Splash* zur Auswertung von JavaScript erweitert werden. Allerdings zeigt sich, dass QT eingesetzt wird und aus diesem Grund keine eigenständige Variante darstellen kann[14].

Auswahl

Weil *PhantomJS* und *QT Webkit* auf der gleichen HTML-Rendering-Engine, auf Webkit[15] basieren (vgl. dazu Abschnitt 2.6), sind keine Unterschiede in den Messergebnissen zu erwarten. Folglich fällt die Wahl auf das *QT Webkit*. Im Folgenden wird gezeigt, wie der darin integrierte Browser modifiziert werden kann, um die geforderten Anforderungen aus Abschnitt 4.5 zu erfüllen.

5.4.2 *Modifikation der Netzwerkkomponente*

Anforderungen

In Abschnitt 5.3.2 wurden verschiedene Anfragetypen beschrieben, die bei der Verarbeitung von Archivwebseiten auftreten können. Diese sind innerhalb der Implementierung zu berücksichtigen. Algorithmus 1 ist auf die Gegebenheiten der gewählten Applikation aus Abschnitt 5.4.1 zugeschnitten. Dabei wird eine Methode der Klasse *QNetworkAccessManager* überschrieben und nimmt fortan die Bewertung vor, ob eine Netzwerkverbindung zu einem Server aufgebaut werden darf oder nicht.

Anfragetypen

Dabei werden die Anfragetypen I bis VI aus Tabelle 5.1 berücksichtigt. Übergeben wird eine Archivwebseite $wURL$. Diese wird angefragt, wenn eine Ressource $sURL$ vom Netzwerkmanager der Programmbibliothek aufgerufen werden soll. Wenn es sich um eine externe Komponente handelt, wird diese als solche in einer Liste der Drittparteien gespeichert. Darüber hinaus wird geprüft, ob der Verweis im Archiv verbleibt oder sich an das „Live Web“ richtet. Sofern letzteres eintritt, muss die Anfrage blockiert werden (Abschnitt 4.4.3). Die Anfrage muss auch dann blockiert werden, wenn

12 PhantomJS onResourceRequested - http://phantomjs.org/api/webpage/handler/on-resource-requested.html, abgerufen am 16.01.2017.

13 PyQt4 - QNetworkAccessManager Class Reference http://pyqt.sourceforge.net/Docs/PyQt4/qnetworkaccessmanager.html, abgerufen am 16.01.2017.

14 Richard Dowinton - Handling JavaScript in Scrapy with Splash https://blog.scrapinghub.com/2015/03/02/handling-javascript-in-scrapy-with-splash/, abgerufen am 16.01.2017.

15 https://webkit.org/, abgerufen am 06.11.2017.

das Jahr der Ressource nicht mit dem Jahr der angefragten Webseite übereinstimmt oder in der Zukunft liegt (Abschnitt 4.4.3).

Input : Anfrage für $rURL$ vom Parser während der Interpretation der Archivwebseite $wURL$.
Output : Erweiterung der Liste $ressourceList$ um externe Aufrufe, die durch Abruf und Analyse der Adresse $wURL$ erzeugt werden.

```
/*                                                    */
/* Überschriebene createRequest() in
   QNetworkAccessManager                              */
/*                                                    */
if sld(rURL) ≠ sld(wURL) then
  | ressourceList[] ← host(rURL) ;       /* Drittpartei. */
end
websiteYear ← parseArchiveYear(wURL) ;
if host(rURL) = *.archive.org then
  | requestYear ← parseArchiveYear(rURL) ;
  | if requestYear ≤ websiteYear) then
  |   | return parent.createRequest(rURL);     /* zulässig */
  | end
end
return parent.createRequest(null);            /* blockieren */
```

Algorithmus 1 : Anpassung der NetworkAccessManager-Klasse.

5.4.3 *Test und Fehlerbehandlung*

Tests

Wie in der Anwendungsentwicklung üblich ist, wurde in diesem Fall die Applikation anhand verschiedener Testfälle auf Funktionalität geprüft. Die Prüfung umfasst Tests zur korrekten Verarbeitung von Archivwebseiten, als auch zur Robustheit gegenüber Fehler, die während einer Messung (bzw. Seitenabfrage) auftreten können.

Fehlerarten und deren Behandlung

Durch die in Abschnitt 5.3.3 beschriebene Verkapselung von Messvorgängen in eigene Prozesse wird die Robustheit des Gesamtsystems verbessert. Im Folgenden werden typische Problemfälle aufgelistet, die bei der Verarbeitung auftreten können und den Messvorgang nicht beeinflussen dürfen:

- Unerreichbarkeit der Archivwebseite,
- keine Rückmeldung vom Prozess,
- Rückmeldung von Fehlermeldungen während des Aufbaus der Webseite und
- unerwartete Beendigung eines Prozesses.

Umgang mit Fehlern

Tritt einer dieser Fälle ein, wird das Ergebnis der Messung als fehlerhaft markiert. Der Umgang mit fehlerhaften Messwerten, während der Durchführung der Messung auftreten, wird in Abschnitt 6.1.2 beschrieben.

Anforderungen

Wie in Abschnitt 4.2 erläutert, wurden in der Designphase Anforderungen erarbeitet, die eine Implementierung berücksichtigen muss. Diese Anforderungen sind in Abschnitt 4.5 des vorherigen Kapitels aufgezeigt worden. Im Folgenden wird betrachtet, inwieweit diese Anforderungen in der Implementierung berücksichtigt wurden.

Prüfung der Anforderungen

Die Prüfung der Anforderungen stellt keine Evaluation des Werkzeugs oder der Studie dar. Vielmehr wird geprüft, ob keine ungewollten Abweichungen zwischen Design, Entwurf und Implementierung entstehen. Eine vollständige Evaluation findet nach Durchführung der Studie in Abschnitt 6.3 statt.

(A-1) **Vollständigkeit**. Diese Anforderung soll sicherstellen, dass die Wahl der Methodik für alle Elemente der Testmenge und für alle Jahrgänge vergleichbare Ergebnisse erzielt. Eine Einschränkung der Messung von Web-Tracking auf Archivwebseiten wurde in Abschnitt 4.4.3 erläutert und innerhalb der Implementierung in den Abschnitten 5.2.3 und 5.3.2 berücksichtigt.

(A-2) **Kontaminationsfreiheit**. Um diese Anforderung zu erfüllen, werden verschiedene Abfrageklassen identifiziert (Abschnitt 5.3.2) und eine Netzwerkkomponente angepasst wie in Abschnit 5.4.2 beschrieben.

(A-3) **Effizienz**. In Abschnitt 5.3.1 wurde unter Berücksichtigung der verfügbaren Werkzeuge eine Werkzeugklasse gewählt, die eine effiziente Verarbeitung ermöglicht. Durch die in Abschnitt 5.3.3 beschriebene Parallelisierung kann darüber hinaus eine Leistungssteigerung bewirkt werden.

(A-4) **Auswertbarkeit**. Abschnitt 5.3.4 beschreib die persistente Speicherung der erhobenen Daten. Die Verwendung einer standardisierten Datenbank vereinfacht die Auswertung durch SQL-Abfragen sowie die Weitergabe für weitere Forschungszwecke.

6

UMSETZUNG UND EVALUATION DER RETROSPEKTIVEN ANALYSE

Zusammenfassung: In diesem Kapitel werden die Durchführung und Ergebnisse der retrospektiven Analyse dokumentiert und präsentiert. Ebenso werden die Ergebnisse u. a. auf Basis vergleichbarer Analysen evaluiert, wobei Gemeinsamkeiten und Unterschiede herausgearbeitet werden. Abschließend findet eine Diskussion der Ergebnisse und ein Ausblick statt.

6.1 DURCHFÜHRUNG DER ANALYSE

Kapitelübersicht

Die Durchführung unterteilt sich in drei Schritte. Im ersten Schritt wird eine Testmenge gewählt. Das verfügbare Material wurde bereits in Abschnitt 4.3 gesichtet und beschrieben. Die gewählten Webseiten sind nicht zwangsweise im Laufe der Jahre archiviert worden. Aus diesem Grund muss für jedes Element der Testmenge geprüft werden, für welche Jahre ein archiviertes Exemplar vorliegt. Sobald innerhalb der Testmenge unter Berücksichtigung der Verfügbarkeit eine Auswahl getroffen wurde, werden die Exemplare im zweiten Schritt vom Archivsystem angefragt und analysiert. Die Messergebnisse werden im dritten Schritt verarbeitet, in eine persistente Speicherung überführt und zur Präsentation vorbereitet.

6.1.1 *Testmenge*

Auswahl der Testmenge

Datenquellen

In Abschnitt 4.3 wurden bereits Datenquellen vorgestellt, die eine Übersicht von häufig besuchte und bekannte Webseiten bieten. Bei korrekten Daten sind diese Webseiten in der Lage, umfangreiche Nutzungsinformationen zu erfassen. Diese Eigenschaft macht sie für Tracker besonders interessant.

Alexa Internet

Alexa Internet ist als Datenquelle für diese Analyse besonders von Interesse, da diese vom *Internet Archive* als Quelle verwendet wird. Darüber hinaus dient Alexa Internet auch in anderen Publikationen [175, 107, 132, 52] als Quelle von Testmengen. Es ist zu beachten, dass sich Auflistung und Sortierung der Alexa-Liste nach aktuellen Popularitätswerten richtet. Dies erschwert eine retrospektive Analyse, da unklar ist, ob die gewählten Webseiten auch in den vergangenen Jahren populär waren. Bei einer Anfrage teilte das Unternehmen mit, dass es keine Daten mehr aus den Jahren vor 2007 gibt (vgl. Anhang E.1). Aus diesem Grund richtet sich die Analyse nach den aktuellen Popularitätswerten. Es wird betrachtet, wie sich die als heute populär bewerteten Webseiten im Verlauf der Jahre verändert haben.

Prüfung auf Verfügbarkeit im Archiv

Verfügbarkeit von Archivdaten

Auch wenn eine Webseite als populär eingeschätzt wird, kann der Betreiber der Speicherung innerhalb des Archivs widersprechen und diese entfernen lassen. Für Zeiträume, in denen Speicherplatz kostenintensiver war als heute, muss davon ausgegangen werden, dass für manche Webseiten keine älteren Aufzeichnungen mehr vorliegen. Im ersten Schritt wurden die ersten 30 000 Einträge in der Alexa-Liste auf Verfügbarkeit im Archiv geprüft. Angefordert wurde das nächste verfügbare Exemplar vom 01. Juli des jeweiligen Jahres.

Zu diesem Zweck bietet der Archivbetreiber ein (REST) Interface; wie beispielsweise zur Abfrage von heise.de:

```
1 http://archive.org/wayback/available?url=heise.de&timestamp
      =20160701
```

Ergebnisse

Die Ergebnisse der Verfügbarkeitsabfragen werden in eine Datenbank übertragen. Das SQL-Statement in Quelltext 6.1 bezieht sich auf die in Abbildung 5.2 gezeigte Datenbankstruktur. Es zählt wie viele Webseiten im Archiv zur Verfügung stehen. Die Ergebnisse der Abfragen können in Tabelle 6.1 eingesehen werden. Es zeigt sich, dass für mehr als 12 000 Webseiten eine Historie von mindestens 10 Jahren zur Verfügung steht. Dies entspricht ca. 1/3 der insgesamt betrachteten Webseiten. Wird die Grenze der notwendigen Jahre auf 10 gesetzt, besteht der Testsatz aus 12 547 Webseiten der Top 30 000 Webseiten.

```
1 SELECT Count(*)
2 FROM   (SELECT websiteid,
3                Count(*) AS c
4         FROM   archive_url
5         WHERE  archive_url <> ''
6         GROUP  BY websiteid)
7 WHERE  c >= ANZ
```

Quelltext 6.1: Verfügbarkeit der archivierten Daten; *ANZ* für die Anzahl der Jahre (hier: 5, 10, 16).

Kriterium	#
Betrachtete Anzahl an Webseiten	30 000
keine Archivdaten vorliegend:	2 615
Verbleibende Webseiten	27 385
davon mehr als (>=) 5 Jahre Historie	21 034
mehr als (>=) 10 Jahren Historie	12 547
mit voller 16-Jahres Historie	3 558

Tabelle 6.1: Anzahl verfügbarer Webseiten im Archiv mit einer Mindestanzahl an Jahren.

6.1.2 *Ausführung*

Analyse der Webseiten

Die Erhebung der Daten wurde zwischen dem 11.04.2016 und 26.04.2016 auf einem Ubuntu 12.04 LTS Betriebssystem durchgeführt. Bei dieser Art der Analyse ist der Erhebungszeitpunkt nicht wesentlich, da davon ausgegangen werden kann, dass die Daten nicht nachträglich verändert werden.

Ergebnis der Ausführung

Bei 5 932 der insgesamt 169 410 abgerufenen Archivwebseiten sind Fehler aufgetreten. Prinzipiell ist es möglich, mit den verbleibenden Daten fortzufahren. In diesem Fall wurden nur die Webauftritte in die Analyse einbezogen, für die keine Fehler aufgetreten sind. Dies bedeutet, dass alle Ergebnisse stets eine Zehnjahreshistorie besitzen und während der Messung keine Fehler aufgetreten sind. Dieser vergleichsweise strenge Umgang mit Fehlern wird aufgrund der Vollständigkeitsanforderung (A-1) gewählt. Infolgedessen bleiben 554 Webseiten von den 12 547 unberücksichtigt: 11 993 Webseiten verbleiben.

6.1.3 *Verarbeitung*

Ziel

Ziel der Analyse ist das Betrachten der Drittparteieinbettungen auf Archivwebseiten. Nach Verarbeitung der Webseiten liegt für jede Webseite der Testmenge, für jedes (verfügbare) Jahr und für jeden Aufruf einer Drittpartei mindestens ein Datensatz vor.

Normalisierung der Referenzen

Kategorien

In Abschnitt 5.3.2 wurde gezeigt, dass es verschiedene Kategorien (I - IV) von Einbettungen und Referenzen gibt. Zum Auswerten der Daten muss dies berücksichtigt werden.

BEISPIEL: GOOGLE-ANALYTICS. Eine Webseite bindet den Dienst google-analytics.com im Webauftritt ein. Während des Archivierungsvorgangs wird diese Referenz erkannt und entsprechend auf das Archiv umgebogen (*archive.org/web/.../google-analytics.com/*). Auf der gleichen Webseite wird an anderer Stelle nochmals eine Einbettung von google-analytics.com vorgenommen, die jedoch nicht vom Crawler erkannt wurde. Demzufolge findet eine Einbettung des Typs III, sowie eine des Typs VI (gemäß Tabelle 5.1) statt.

Normalisierung

Um diese verschiedenen Formen der Einbettung zu normalisieren, werden in der Tabelle *request* die Attribute *original_host* und *original_domain* gesetzt. In diesem Fall steht das Original nicht für den tatsächlich gemessenen Wert, sondern für den Wert, den der Betreiber ursprünglich innerhalb der Webseite benutzt hat.

Datenreduktion

Vereinfachung

In Abbildung 5.2 wurde die Struktur der Datenbank dargestellt. Die Tabelle

request enthält Einträge für jeden Aufruf einer externen Ressource. In der Auswertung ist jedoch die Anzahl der Aufrufe einer externen Ressource für eine Webseite ohne Bedeutung, da bereits der erste Aufruf als kritisch anzusehen ist. Infolgedessen ist eine Reduzierung der Daten möglich, die über die Hilfstabelle *request_summary* umgesetzt wird. Diese enthält für jeden Analyseprozess alle aufgerufenen Domains jeweils einmal.

Ergebnis

Die resultierende Datenbankstruktur ist in Abbildung A.2 im Anhang dargestellt. Diese vereinfacht Auswertungen, die in der folgenden Auswertung präsentiert werden.

Auswertungen

Einbettungen

ANZAHL DER EINBETTUNGEN Eine Übersicht für die Einbettungen pro Webseite pro Jahr kann mit einer geeigneten Abfrage erzeugt werden wie in Quelltext 6.2 dargestellt ist. Diese ist besonders für das Ermitteln der statistischen Kennzahlen wie Erwartungswert und Varianz wichtig. Das Ergebnis dieser Abfrage ist eine Liste der betrachteten Webseiten und die Anzahl der unterschiedlichen Drittparteieinbettungen.

```
1   SELECT process.id                                    AS
        pid,
2          website.url                                   AS
               url,
3          archive_url.year                              AS
               year,
4          (SELECT Count(*) AS a
5           FROM   request_summary
6           WHERE  request_summary.processid = process.id) AS
                cons
7   FROM   process
8          LEFT JOIN archive_url
9                  ON ( process.archive_urlid = archive_url.id
                        )
10         LEFT JOIN website
11                 ON ( archive_url.websiteid = website.id )
12  WHERE  ( status = 2 AND result = 0 )
13         AND archive_url.year = RYEAR
```

Quelltext 6.2: Abfrage der Anzahl der Drittparteieinbettungen für das jeweilige Jahr (RYEAR).

Topliste

HÄUFIG EINGEBUNDENE DRITTPARTEIEN Eine Übersicht über die Häufigkeit der Einbettung von Drittparteien kann über den Quelltext 6.3 bezogen werden. Daraus ergibt sich eine Rangliste von Drittparteien für das angefragte Jahr.

```
SELECT original_domain,
       Count(*) AS cc
FROM   request_summary
       LEFT JOIN process
           ON ( request_summary.processid = process.id )
       LEFT JOIN archive_url
           ON ( process.archive_urlid = archive_url.id )
WHERE  archive_url.year = RYEAR
GROUP  BY original_domain
ORDER  BY cc DESC
```

Quelltext 6.3: Rangliste der am häufigsten eingebetteten Drittparteien für das jeweilige Jahr (RYEAR).

Coverage

ABDECKUNG Die Abdeckung (Coverage) ist das Verhältnis aller betrachteten Webseiten zu den Webseiten, die durch die ersten X-(Achse)-Plätze der häufigst eingebetteten Drittparteien abgedeckt werden. Zusätzlich kann ermittelt werden, wie hoch die Wahrscheinlichkeit ist, auf einer Webseite einen Tracker aus den ersten Top X vorzufinden. In Anhang C ist in Quelltext C.6 diese Form der Abfrage dargestellt.

Graph

NETZWERKGRAPH Die gemessenen Daten ermöglichen die Generierung eines Graphen. Dabei stellen die überprüften Webseiten sowie die eingebundenen Drittparteien die Knoten dar. Die Kanten werden ab mindestens einer Abfrage gesetzt. So können häufig auftretende Drittparteien gesondert markiert werden, um Zu- oder Abnahme im Laufe der Jahre beobachten zu können.

Visualisierung des Netzwerks

Zu diesem Zweck wurde die Programmbibliothek NetworkX zur Generierung, Analyse und Darstellung von Graphen verwendet [70], die bereits häufig in wissenschaftlichen Veröffentlichungen[1] eingesetzt wurden [4, 25, 26].

TLDs

TOP-LEVEL-DOMAINS Für die Unterteilung der Ergebnisse nach Top-Level-Domain (TLD) müssen im ersten Schritt alle vertretenen TLDs ermittelt werden. Eine reine Betrachtung des Suffix der Domain genügt nicht, weil zusammengesetzte Domains (z. B. *.co.uk*) existieren. Aus diesem Grund wird die vom Mozilla-Projekt gepflegte „Public Suffix List“[2] verwendet.

Unterscheidung nach Regionen

Anschließend erfolgt eine Auswertung der Drittparteieinbettungen pro Top-Level-Domain und wird durch die Abfrage in Quelltext 6.4 ermöglicht. Die Verbindungsdaten werden analog zu Abschnitt 6.1.3 analysiert.

```
SELECT (SELECT Count(*) AS a
        FROM   request_summary
        WHERE  request_summary.processid = process.id) AS cons
FROM   process
       JOIN archive_url
```

1 Nach eigenen Angaben gemäß http://networkx.readthedocs.io/en/networkx-1.11/bibliography.html, abgerufen am 30.01.2017.

2 https://publicsuffix.org/list/public_suffix_list.dat, abgerufen am 31.01.2017.

```
6           ON ( process.archive_urlid = archive_url.id )
7         JOIN website
8           ON ( website.id = archive_url.websiteid )
9              AND process.result = 0
10             AND archive_url.year = RYEAR
11             AND website.url LIKE RURL
```

Quelltext 6.4: Abfrage der Einbettungen eines Adressmusters (RURL) für das jeweilige Jahr (RYEAR).

6.2 Präsentation der Ergebnisse

Analysen

Die Auswertungen zur Präsentation der Ergebnisse orientieren sich an Erhebungen aus bestehenden Arbeiten (Abschnitt 3.4). Ziel sind vergleichbare Analysen, um auf Gemeinsamkeiten und Unterschiede innerhalb der Evaluation eingehen zu können.

6.2.1 *Allgemeine Übersicht*

Topliste

In Tabelle 6.2 ist die Anzahl der verfügbaren Webseiten für das jeweilige Jahr zu sehen. Für das Jahr 2000 liegen nur 6506 Webseiten vor; für das Jahr 2011 stehen die meisten Webseiten zur Verfügung: 11.210. Es ist zu erkennen, dass die Testmenge je weiter man zurück geht, immer mehr ausdünnt. Aus diesem Grund wurde davon abgesehen, Webseiten vor dem Jahr 2000 (bis zur theoretisch möglichen Rückblickgrenze des Jahres 1996) in die Analyse aufzunehmen.

6.2.2 *Statistische Kennzahlen*

Statistische Auswertungen

Zunächst werden statistische Kennzahlen aus den Daten ermittelt werden. Über den bereits vorgestellten Quelltext 6.2 lassen sich Webseiten und deren Einbettungen ermitteln, die anschließend zu den Daten in Tabelle 6.2 weiterverarbeitet werden können. Ausgangsbasis der Analyse ist die (aufsteigend sortierte) geordnete Menge n-tupel $N = (x_1, x_2, \ldots, x_n)$.

Definition der Kennzahlen

$Jahr$: Das betrachtete Jahr (in einem Zeitraum von 2000 bis 2015).

\# : Die Anzahl der zur Verfügung stehenden Webseiten.

x_{min} : Das Minimum, der geringste aufgetretene Wert: $\forall x \in N : x_{min} \leq x$

$\bar{x}_{med}$: Der Median: $\bar{x}_{med} = \begin{cases} x_{\frac{n-1}{2}} & n \text{ ungerade} \\ \frac{1}{2}\left(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}\right) & n \text{ gerade.} \end{cases}$

Der Median kann auch als zweites Quartil $x_{Q0.50}$ betrachtet werden.

$x_{Q0.25}$: Das untere Quartil; analog dem Median, allerdings wird nicht der Wert bei der Hälfte betrachtet, sondern nach dem ersten Viertel.

$x_{Q0.75}$: Das obere Quartil; analog dem Median mit Betrachtung des Wertes nach 75 % der Werte.

x_{max} : Maximal gemessener Wert: $\forall x \in N : x \leq x_{max}$

$\bar{x}$: Das arithmetische Mittel (Durchschnitt) $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.

σ : Die Standardabweichung der Grundgesamtheit

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2}.$$

$Var(x)$: Die Varianz $\sigma^2 = Var(X)$.

NumPy

Zur Berechnung wurden die Statistikfunktionen `min`, `max`, `mean`, `percentile`, `median`, `std` und `var` der Programmbibliothek NumPi[3] verwendet.

Jahr	#	x_{min}	$x_{Q0.25}$	$\tilde{x}_{med}$	$x_{Q0.75}$	x_{max}	$\bar{x}$	σ	$Var(x)$
2000	6504	0	0	0	1	18	0,61	1,31	1,72
2001	7741	0	0	0	1	24	0,77	1,55	2,39
2002	8314	0	0	0	1	33	0,81	1,56	2,43
2003	8923	0	0	0	1	34	0,85	1,62	2,61
2004	9783	0	0	0	1	36	0,99	1,89	3,56
2005	10318	0	0	0	1	44	1,13	2,00	4,00
2006	10817	0	0	1	2	49	1,45	2,27	5,17
2007	11013	0	0	1	3	44	1,92	2,54	6,44
2008	11346	0	0	1	3	35	2,09	2,57	6,58
2009	10899	0	1	2	4	38	2,53	2,86	8,21
2010	10964	0	1	2	4	46	3,07	3,25	10,58
2011	11208	0	1	3	6	44	3,91	3,90	15,24
2012	11174	0	1	4	7	44	4,63	4,33	18,77
2013	11174	0	2	4	8	44	5,44	4,92	24,20
2014	10860	0	2	5	9	60	6,27	5,30	28,12
2015	10736	0	2	5	10	57	6,61	5,50	30,28

Tabelle 6.2: Statistische Kennzahlen für Webseiten mit einer Historie von mindestens 10 Jahren: Jahr, Anzahl Minimum, unteres Quartil, Median, oberes Quartil, Maximum, arithmetische Mittel, Standardabweichung, Varianz.

Box-Whisker-Plot

Einen Teil der Kennzahlen lassen sich in einem Box-Whisker-Plot darstellen, wie Abbildung 6.1 zeigt. Zu beachten ist, dass die Whisker (Antennen) grundsätzlich die minimalen und maximalen Werte darstellen, jedoch üblicherweise die Länge gedeckelt ist. In diesem Fall wurde diese Länge auf 1,5*IQR (=Interquartilsabstand, IQR $=_{def}$ $x_{Q0.75} - x_{Q0.25}$) beschränkt, wie bei Box-Whisker-Plots allgemein üblich ist.

6.2.3 *Rangliste der Drittparteien*

Rangliste

Basis dieser Auswertung ist die Abfrage 6.3: diese liefert eine sortierte Liste über Drittparteien und die Anzahl der Einbettungen. Dabei wird jede Webseite nur einmal gezählt, so ist die Häufigkeit der Einbettungen derselben Drittpartei unerheblich.

3 Statistics - NumPy v1.12 https://docs.scipy.org/doc/numpy/reference/routines.statistics.html, abgerufen am 27.01.2017.

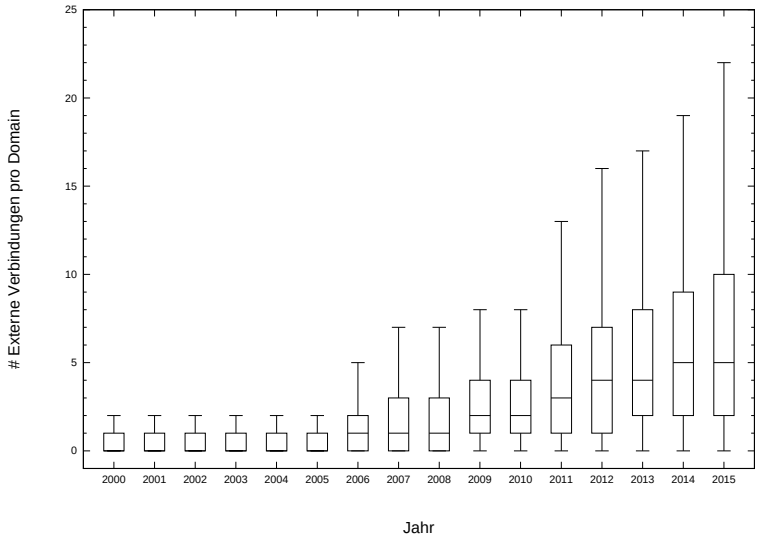

Abbildung 6.1: Box-Whisker-Plot (ohne Ausreißer) zur Visualisierung von Tabelle 6.2.

Die 15 am häufigsten eingebetteten Drittparteien eines jeden Jahres können in Tabelle 6.3 betrachtet werden.

6.2.4 *Kummulative Abdeckungsanalyse*

Abdeckung der Tracker

Die in Abschnitt 6.1.3 erläuterte Abdeckung (Coverage) kann in Abbildung 6.2 betrachtet werden. Zum Verständnis des Graphen: Bei $x = 1$ ist die Abdeckung des am häufigsten eingesetzten Trackers zu sehen. Bei $x = 2$ ist die Abdeckung von Platz 1 und Platz 2 zusammen ablesbar bis $x = 50$, bei der die ersten 50 Plätze für die Jahre 2000, 2005, 2008, 2010 und 2015 zusammen betrachtet und erhoben werden. Diese Analyse ermöglicht zu ermitteln, welche Menge an Webseiten von den Top 1-50 der häufig eingebundenen Drittparteien erfasst werden.

6.2.5 *Netzwerkgraph*

Graph

Die ermittelten Verbindungsdaten ermöglichen den Aufbau eines Netzwerkgraphen. In den Abbildungen 6.3 bis 6.8 sind die betrachteten Webseiten sowie eingebetteten Drittparteien als Knoten dargestellt. Aus Gründen der Übersicht werden hier nur die Kanten gezeichnet. Die jeweils 5 stärksten Tracker sind mit einem Buchstaben markiert und können in Tabelle 6.3 eingesehen werden.

6.2.6 *Analyse der Top-Level-Domain*

TLDs

Die fünf häufigsten Top-Level-Domains der ausgesuchten Testmenge wurden auf die Anzahl ihrer Einbettungen betrachtet. Das in Tabelle 6.4 abge-

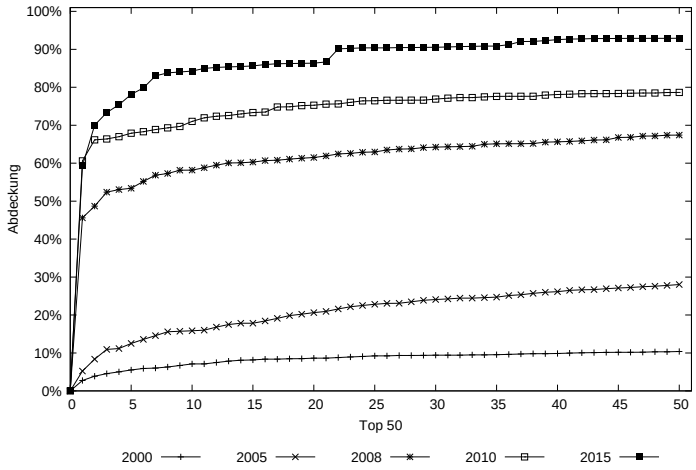

Abbildung 6.2: Abdeckungsanalyse der Top 50 Tracker.

bildete Ergebnis zeigt die Anzahl der Webseiten mit den TL-Domains com, net, ru, org und de. Daraus lässt sich der Anteil an der Gesamtmenge (zweite Spalte) erschließen. Zum Vergleich ist für jedes Jahr, für jede TLD das arithmetische Mittel für die Anzahl der Einbettungen zu sehen, wie es bereits in Abschnitt 6.2.2 für Tabelle 6.2 definiert wurde.

Unterscheidung nach Regionen

Die Top-Level-Domain ist kein „hartes“ Unterscheidungskriterium für die Herkunft des Betreibers oder für die Zielgruppe einer Webseite. Eine Webseite für den deutschen Markt kann auch unter .net oder .com zu finden sein. So wie die Zugehörigkeit einer Webseite zu einer Person oder einem Unternehmen nicht stets feststellbar ist, ist auch eine Unterscheidung des jeweiligen Landes nicht möglich. Prinzipiell könnte die IP-Adresse zwar einen Hinweis auf den Serverstandort liefern, allerdings liegen in den archivierten Webseiten keine IP-Adressinformationen zum Zeitpunkt der Erfassung vor.

6.3 EVALUIERUNG

Die Evaluation unterteilt sich in drei Teile:

MESSWERKZEUG. Das zur Studie verwendete Messwerkzeug muss auf seine Verlässlichkeit geprüft werden. Neben den innerhalb der Softwareentwicklung durchgeführten Tests muss darüber hinaus verglichen werden, ob die Ergebnisse auch bei Einsatz von anderen Werkzeugen reproduzierbar sind.

STUDIE. Die Studie selbst wird auf zwei verschiedene Arten evaluiert:

VERGLEICH ARCHIVE- ZU LIVE WEB. In Abschnitt 4.4.3 wurden die Grenzen einer retrospektiven Analyse aufgezeigt. Insbesondere in Bezug auf die Verweise ins Live Web soll erneut geprüft und getestet werden, wie verlässlich die Informationen aus dem Archivbestand sind.

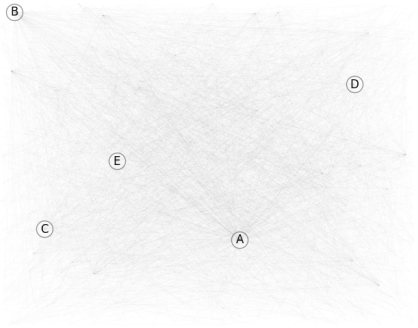

Abbildung 6.3: Jahr 2000

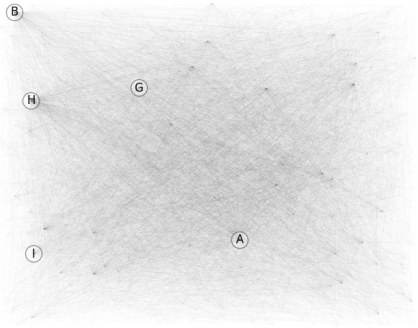

Abbildung 6.4: Jahr 2005

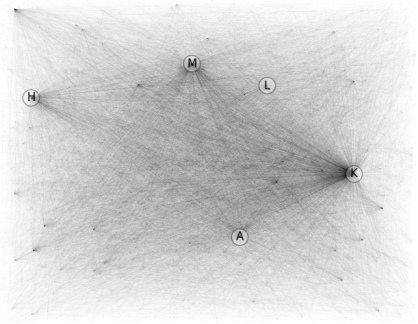

Abbildung 6.5: Jahr 2007

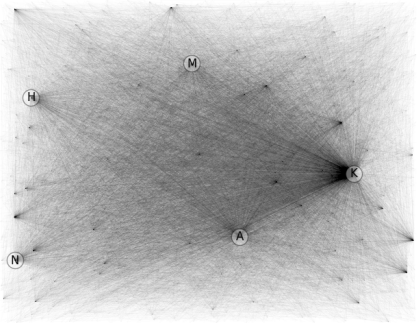

Abbildung 6.6: Jahr 2010

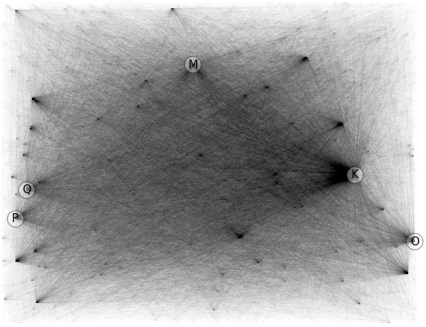

Abbildung 6.7: Jahr 2012

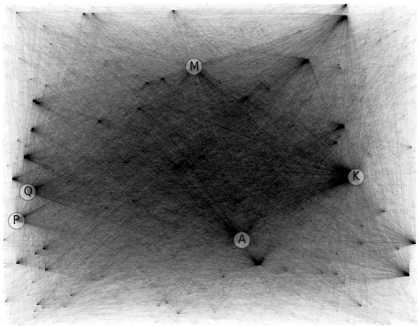

Abbildung 6.8: Jahr 2015

VERGLEICH ZU BESTEHENDEN STUDIEN. Der Vergleich der Ergebnisse mit bestehenden quantitativen Erhebungen stellt den Kern der Evaluation dar. Dabei werden Abweichungen sowie Gemeinsamkeiten herausgearbeitet und diskutiert.

In Abschnitt 6.3.2 wird ein Vergleich zur Studie von Lerner et al. durchgeführt und in Abschnitt 6.3.3 ein Fazit zur Evaluation gezogen.

6.3.1 *Evaluation des Messwerkzeugs*

Werkzeugwahl

Die Gründe für die Wahl des Messwerkzeugs wurden in Abschnitt 5.3.1 dargelegt. Es muss sichergestellt sein, dass die Applikation korrekte Ergebnisse liefert. Eine teilweise Evaluation der Funktionsweise des Werkzeugs findet durch die Durchführung geeigneter Testfälle statt, wie in Abschnitt 5.4.3 beschrieben ist.

Evaluierungswege

In diesem Abschnitt wird ein weiterer Evaluationsansatz vorgestellt. Geprüft wird, ob die erzeugten Ergebnisse mit vergleichbaren Werkzeugen erzielt worden wären. Es wurde eine zufällige Testmenge von 50 Adressen aus der Gesamtmenge von 161774 archivierten Webseiten ausgewählt und mit den Ergebnissen einer manuellen Analyse verglichen, die ein Mozilla Firefox Browser in der Version 55.0.2 und ein Google Chrome Browser in der Version 61.0.3163.100 erzeugen. Dafür wurden die integrierten Entwicklerwerkzeuge der Browser verwendet, die eine Auflistung der aufgerufenen Netzwerkadressen ermöglichen. Die Konfigurationen des Browsers wurden nicht verändert.

```
1  SELECT *
2  FROM   archive_url
3         LEFT JOIN process
4                ON ( process.archive_urlid = archive_url.id )
5  WHERE  status = 2
6  ORDER  BY Random()
7  LIMIT  50
```

Quelltext 6.5: Zufällige Auswahl einer Testmenge an archivierten Webseiten aus der Datenbank.

Testmenge

Die zufällig gewählten Webseiten sind im Anhang D.1 zu finden. Die Untersuchung hat gezeigt, dass es keine Unterschiede zwischen dem Browser und Werkzeug gibt. Da es sich um archivierte Inhalte handelt, kann die Messung reproduziert[4] werden.

6.3.2 *Evaluation der Studie*

Evaluierung der Messdaten

Neben der Evaluation des Werkzeugs werden im Folgenden die Ergebnisse der Studie evaluiert. Dies umfasst im ersten Schritt den Vergleich, welche

4 Es muss beachtet werden, dass Inhalte auch zufällig generiert und sich aus diesem Grund ändern können. Dies wurde bei der genannten Testmenge nicht festgestellt.

Unterschiede sich aus der Messung einer archivierten Webseite und einer Messung im „Live Web“ ergeben. Obwohl dies nur beispielhaft gezeigt wird, kann auf diese Weise eine Einschätzung der Messungenauigkeit vermittelt werden. Im zweiten Schritt wird ein Vergleich der Studienergebnisse mit quantitativen Arbeiten durchgefügt, wobei auf Gemeinsamkeiten und Unterschiede eingegangen wird.

Vergleich Live Web zu Archived Web

Abweichungen

In Abschnitt 4.4.3 wurde beschrieben, wie Verweise ins „Live Web“ zwar gemessen, jedoch zur Gewährleistung der (A-2) Kontaminationsfreiheit nicht verfolgt werden können. Dieses Problem ist in Abbildung 6.9 grafisch dargestellt. Das Beispiel I. zeigt eine Erstpartei einer archivierten Webseite, die auf Drittparteien im Live Web und ins archivierte Web verweist. Im Vergleich zu II. ist zu erkennen, dass Aufrufe von Ressourcen außerhalb des archivierten Inhalts nicht berücksichtigt werden können.

Beispiel

Dieses Problem wird an einem konkreten Beispiel verdeutlicht. Die Webseite http://www.spiegel.de wurde am 06.10.2017 um 14:50 Uhr vermessen und mit den Ergebnissen der archivierten Version[5] der Webseite von 12:21 Uhr des gleichen Tages verglichen. Die Ergebnisse sind in Tabelle 6.5 zusammengefasst.

Unerkannte Tracker

Der Tracker von `doubleclick.net` wurde in der archivierten Version nicht gefunden bzw. ein Aufruf zu entsprechenden Domain wurde nicht ausgelöst. In diesem konkreten Fall ist der Quelltext der Webseite genauer zu betrachten, um eine Erklärung für diesen Umstand zu gewinnen. Aufgrund der umfangreichen Obfuskierungsmaßnahmen durch den Betreiber sind die Einflussfaktoren nicht nachvollziehbar. Um auszuschließen, dass dies durch den User-Agent oder das Messwerkzeug bedingt ist, wurde die archivierte Webseite mit weiteren (wie in Abschnitt 6.3.1 beschrieben) Browsern geöffnet. Auch in diesen Tests wurde keine Verbindung zu DoubleClick ausgelöst.

Gründe

Unter Berücksichtigung dieses Tests ist anzunehmen, dass manche Trackingverfahren nicht mit archivierten Webseiten kompatibel sind bzw. auf diesen nicht entdeckt werden. Die Gründe hierfür können vielseitig sein. Es ist unter anderem möglich, dass das zur Einbettung verantwortliche Script erkennt, dass es sich um eine archivierte Version handelt und aus diesem Grund nicht aktiv wird. Des Weiteren ist eine Beeinflussung des Inhalts durch den Crawler denkbar, der für die Archivierung der Webseite verantwortlich war.

Alternative Lösungen

Zu diesen Zweck kommt eine Vergleichsanalyse der Ergebnisse einer Messung des Live Webs mit der archivierten Version in Frage, sofern für den jeweiligen Tag eine Archivversion zur Verfügung steht. So kann ermittelt werden, wie viele Drittparteien nicht berücksichtigt wurden. Dieses Delta kann anschließend zur Bewertung der Ergebnisse hinzuaddiert werden.

Nachteile

Eine solche Form der Evaluation weist jedoch einige Schwächen auf. Zum

5 https://web.archive.org/web/20171006121722/http://www.spiegel.de/, abgerufen am 12.10.2017.

einen können die Ergebnisse schon bei zwei hintereinander ausgeführten Messungen des „Live Web“ abweichen. Dies ist der Fall, wenn ein von der besuchten Webseite eingebundene Werbepartner zufällig selektiert wird. Zum anderen lässt sich eine solche Vergleichsmessung nur für den aktuellen Zeitpunkt durchführen. Dabei ist unklar, ob dieses Problem im Jahr 2000 (und folgende) stärker oder schwächer ausgeprägt gewesen ist.

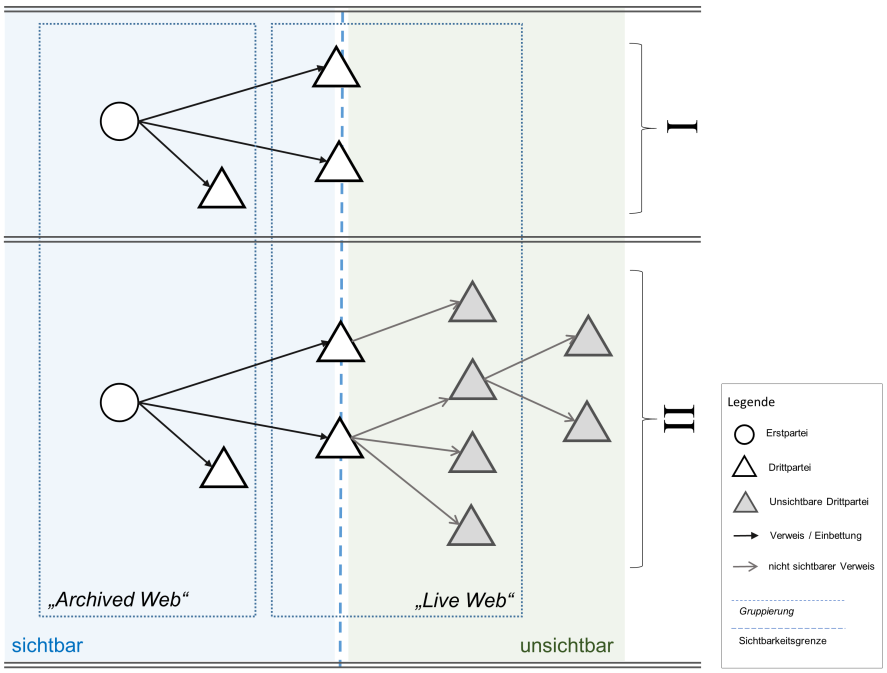

Abbildung 6.9: Vergleich Live Web vs. Archived Web.

Schlussfolgerung

Aufgrund dieser gravierenden Mängel, die an den Ergebnissen zweifeln lassen würden, wurde von einer solchen Untersuchung abgesehen. In Tabelle B.1 im Anhang finden sich die Messergebnisse von zehn Webseiten bestehend aus der Anzahl der Drittparteien, die nur im Live Web, nur im archivierten Web oder in beiden gefunden wurden.

Vergleich der Ergebnisse mit bestehenden Arbeiten

Vergleich mit vier Studien

Im Folgenden werden die Ergebnisse der retrospektiven Analyse mit bestehenden quantitativen Analysen verglichen. Die Auswahl der Studien, die mit den hier erzielten Ergebnissen verglichen werden, ergibt sich aus Abschnitt 3.4. Geeignet sind Studien, die einen allgemeinen Fokus haben bzw. deren Ergebnisse nicht auf spezielle Techniken oder Zielgruppen beschränkt sind.

Folgende Studien werden berücksichtigt:

- Krishnamurthy und Wills (2009),
- Chaabane et al. (2012),
- Libert (2015) und
- Engelhardt und Narayanan (2016).

Die Vergleichszahlen, die in Abschnitt 6.2 präsentiert oder durch referenzierte Verfahren erhoben worden sind, stehen in eckigen Klammern (zum Beispiel [5 %]) hinter den Werten der jeweiligen Studie angegeben.

Vergleich mit Krishnamurthy und Wills (2009)

Umfang Krishnamurthy und Wills [96] präsentierten im Jahr 2009 die Ergebnisse von 5 Messungen bzgl. der Ausbreitung von Web-Tracking. Diese wurden im Oktober 2005, April 2006, Oktober 2006, Februar 2008 und September 2008 durchgeführt. Im Folgenden werden die Ergebnisse von 2005, Oktober 2006 und September 2008 mit denen der Jahre 2005, 2006 und 2008 verglichen.

Übereinstimmungen Gute Übereinstimmungen zeigen sich beim Vergleich der Einbettungen von Google-Analytics, die in den Jahren 2005 bei 0 % [0,1 %], 2006 bei 13 % [12,9 %] und 2008 bei 33 % [40,3 %] liegen. Starke Abweichungen hingegen ergeben sich bei Doubleclick, das 2005 mit 16 % [3,3 %], 2006 mit 22 % [3,7 %] und 2008 mit 35 % [7,4 %] gemessen wurde.

Abdeckung Schlüssige Werte ergibt ein Vergleich der Abdeckungsanalyse (Abbildung 6.2) der Top 10 der häufigsten Tracker: 2005 lag diese bei 40 % [15 %], 2006 bei 52 % [29 %] und 2008 bei 70 % [58 %].

Abweichungen Vergleichsweise hohe Werte wurden für Tracker angegeben, die innerhalb der Archivanalyse nur schwach vertreten waren. Dies ist bei atdmt.com und 2mdn.com der Fall. Der Tracker atdmt.com sowie Gründe weshalb dieser in der Analyse nur schwach vertreten sein kann, ist bereits in Abschnitt 4.4.3 näher beschrieben worden. Die Domain 2mdn.com ist zu Doubleclick zugehörig, so dass hier die gleiche Begründung wie bei Doubleclick selbst angenommen werden kann.

Vergleich mit Chaabane et al. (2012)

Ziele Die Studie von Chaabane et al. von 2012 [30] fokussierten die Einbettung sozialer Netzwerke im Web. Während für eine weitere Erhebung in dieser Veröffentlichung eine genaue Zeit angegeben wird, finden sich keine Angaben zum Erhebungszeitpunkt dieser Studie. Weil die Erhebungen und deren Veröffentlichung vermutlich dicht zusammenliegen, wird von einer Erhebung im Jahr 2011 ausgegangen und mit entsprechend diesem Jahr verglichen.

Aufbau Im ersten Teil der Studie steht eine Verbreitungsanalyse von „Online Social Networks“ (OSN): Facebook, Twitter und Google+. Zusätzlich werden auch beliebte Tracker (Google-Analytics, Google AdSense, etc.) gemessen.

Abdeckung Die Studie gibt eine Abdeckung durch Google von 43,29 % [67,5 %] der 9933 analysierten Webseiten an. Es werden allerdings nur ungenaue Angaben darüber gemacht, wie dieser Wert ermittelt wurde. Auch widerspricht dieser überraschend geringe Wert den bereits vorgestellten Ergebnissen von Krishnamurthy und Wills [96] aus dem Jahr 2009, die bereits 2008 eine Abdeckung von nahezu 60 % gemessen hatten:

> „The end result is that in the Sep'08 epoch, the Google family has a reach of nearly 60 % amongst the set of domains in our core test data set – the highest among all third parties by far. "
> Quelle: [96, S. 544]

Soziale Netzwerke

Für Twitter mit 7,25 % [5,6 %] und Facebook 22,2 [18 %] liegen die Ergebnisse im erwarteten Rahmen. Ein Vergleich von Google+ ist zwar technisch möglich, jedoch fragwürdig: Wie in Abschnitt 6.1.1 erklärt ist, wird ein Archivdokument nahe dem 01. Juli des jeweiligen Jahres angefordert. Google+ wurde am 28.06.2011 veröffentlicht[6]. Die im Anhang zu findende Abfrage C.9 kann zum Ermitteln der Anzahl der Einbettungen von Google+ genutzt werden. In der Studie wird dieser Anteil mit 10 % angegeben, während aus den genannten Gründen nur mit 3,6 % gemessen wurde. Für 2012 ergibt sich der Wert 12,6 %, was diese Angaben bestätigt.

Weiteres

Der zweite Teil der Studie behandelt die Analyse von Netzwerkmitschnitten, die hier der Vollständigkeit halber erwähnt sein soll.

Vergleich mit Libert (2015)

Ziele

Die im Jahr 2015 veröffentlichte Arbeit von Timothy Libert (University of Pennsylvania) [107] umfasst die Analyse von 950.489 Webseiten bezüglich der Einbettung von Drittparteien. Durchgeführt wurde diese im Mai 2014, weshalb die Ergebnisse im Folgenden mit den Ergebnissen des Jahres 2014 verglichen werden.

Einbettungen

Die durchschnittliche Anzahl von Einbettungen wurde mit 9,47 [6,27] angegeben. Ein Grund für diesen Unterschied wurde in Abschnitt 6.3.2 verdeutlicht und in Abbildung 6.9 dargestellt. Darüber hinaus muss berücksichtigt werden, dass in der Studie DNS-Informationen genutzt werden, um eine deutlichere Unterscheidung von Erst- zu Drittanbietern vorzunehmen, wie in Abschnitt 4.4.3 beschrieben ist.

Abdeckung

In Tabelle 2 der Studie sind Angaben zu der Abdeckung einzelner Unternehmen. Zunächst scheint ein Unterschied bei der Einbettung von Google-Analytics auffällig zu sein. Die Studie zeigt, dass in 46 % [56 %] der Webseiten ein Trackingpixel (`_utm.gif`) von Google-Analytics geladen wurde. Wie aus den Originaldaten[7] der Libert-Studie entnommen werden kann, finden auch Google-Analytics Einbettungen ohne die Verwendung eines solchen Bildes statt. Ein Beispiel findet man ab Zeile 106 der Originaldaten zur Domain http://0-360.com. Es ist nicht nachvollziehbar, aus welchem Grund der Autor die Analyse auf eingebettete Bilder eingeschränkt hat. Insbesondere da laut Aussage des Autors die am häufigsten nachgeladene Datenart JavaScript-Dateien (36 %) waren. Auf eine eigene Auswertung der Daten von Libert wurde an dieser Stelle verzichtet.

Facebook und Google

Im Folgenden werden nur auf einige Beispielwerte aus dem Text der Ver-

6 https://digiday.com/media/timeline-google-plus-demise/, abgerufen am 12.10.2017.
7 https://timlibert.me/public_data/Alexa_1M_201405-3P_REQUESTS_RAW.csv.gz, abgerufen am 12.10.2017.

öffentlichung zurückgegriffen. Eine Übereinstimmung fand sich bei Twitter mit 17,89 % [14,4 %]. Zum Vergleich des Ergebnisses der Einbettungen von Facebook wurde eine eigene Abfrage für das Jahr 2014 gestartet, die im Anhang im Quelltext C.7 zu sehen ist. Für Facebook wurde ein Gesamtabdeckung von 32,42 % [27 %] festgestellt. Die für Google modifizierte Abfrage C.8 ermöglicht einen Vergleich der angegebenen 78 % [77,6 %] der Webseiten, die einen Aufruf zu einer Google-Domain verursachen.

Abweichungen

Ein starker Unterschied mit 21,36 % [4,4 %] findet sich bei Akamai. Dies liegt darin begründet, dass Akamai auch Inhalte für große Unternehmen wie u. a. Facebook bereitstellt. Weil „Live Web"-Kommunikation unterdrückt wird, findet kein Nachladeprozess von einem solchen Hoster statt. Es ist anzunehmen, dass ein großer Anteil im Facebook-Ergebnis enthalten ist. Eine Betrachtung der Libert-Daten erhärtet diese Vermutung: In den ersten 2000 Zeilen der Ergebnisdatei trat Akamai 47-mal durch die Domain fbstatic-a.akamaihd.net in Erscheinung. Der Name impliziert die Bereitstellung statischer Inhalte für Facebook.

TLDs

Die Studie untersucht die durchschnittliche Anzahl der Einbettungen für verschiedene Top Level Domains wie sie in Abschnitt 6.2.6 durchgeführt wurde. Am stärksten treten dabei die TL-Domains .com mit 10,2 [6,73] und .net mit 10,2 [6,54] Einbettungen (im Durchschnitt) hervor. Für .org zeigte die Studie durchschnittlich 7,81 [4,76] Einbettungen mit einer Differenz von 1,66 [1,52] zum Durchschnitt. In der Libert-Studie liegen .de-Adressen mit 7,52 [6,2] Einbettungen deutlich unterhalb des Durchschnitts.

Vergleich mit Englehardt und Narayanan (2016)

Ziele

In der Studie von Englehardt und Narayanan [48] werden die Ergebnisse einer Messung von 1 000 000 Webseiten vorgestellt, die im Januar 2016 durchgeführt wurde. Die Ergebnisse werden mit dem Jahr 2015 verglichen.

Einbettungen

Gute Übereinstimmungen zeigen sich bei Google-Analytics mit 67 % [55,3 %], wie erwartet starke Abweichungen bei DoubleClick 51 % [25,2 %] (vgl. Abschnitt 6.3.2). Während bei den Werten zu facebook.net gute Übereinstimmung vorlag (25 % [22 %]), wurde bei facebook.com ein deutlicher Unterschied festgestellt (32 % [10 %]). Nimmt man beide Domains zusammen, ergibt sich eine Abdeckung von 27,5 % für das Jahr 2015 (Quelltext C.7), was wiederum gut mit den Ergebnissen von Englehardt und Narayanan vereinbar ist. Ebenfalls gute Übereinstimmung zeigen sich bei twitter.com mit 15 % [14,4 %] und ajax.googleapis.com 22 % [20 %]. Die Abweichungen bei den Domains fonts.googleapis.com 40 % [18,8 %] und gstatic.com 50 % [11,6 %] sind durch blockierte Verweise ins Live Web zu erklären.

Abdeckung

Für Google-Domains wird in [48] eine Abdeckung zwischen 75 % und 83 % angegeben. Diese stimmt mit den Ergebnissen der Abdeckungsanalyse (Abbildung 6.2) überein. Für die ersten vier der häufigsten Drittparteien, die Google angehören, ergibt sich eine Abdeckung von 75 %. Das am zweithäufigsten eingebundene Unternehmen ist Facebook (25 %-33 % [27,5 %].

Darüber hinaus werden in der Publikation weitere Tracking- und Fingerprintingverfahren vorgestellt. *Weiteres*

Abgrenzung zu Lerner et al. (2016)

Im August 2016 wurde eine Studie von Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno und Franziska Roesner veröffentlicht [103], in der eine vergleichbare Messung und Analyse auf Archivdaten durchgeführt wurde. *Ziele*

Im Folgenden werden wesentliche Gemeinsamkeiten und Unterschiede genauer ausgeführt.

UMFANG. Die Studie umfasst eine Analyse von 500 Webseiten, die manuell selektiert wurden. Auch wenn dies sicherlich genügt, um die Technik und um erste Ergebnisse zu demonstrieren, ist ein Vergleich mit Studien schwierig, die 10 000 oder 1 Million Webseiten analysieren.

TOOLING. Die Messung wurde auf Basis einer Browsererweiterung des Chrome Browsers durchgeführt, der mittels externer auf Python-basierte Skripte angesteuert wird. Die Erweiterung ist u. a. das Zulassen und Blockieren von Anfragen verantwortlich und dient der Übertragung der Messergebnisse, die anschließend in einer Datenbank (MongoDB) gespeichert werden. Das Werkzeug wurde für eine vergleichsweise kleine Testmenge entwickelt und lässt keine parallele Verarbeitung zu.

TRACKER-KLASSIFIZIERUNG. Die Autoren unternehmen den Versuch, das durchgeführte Tracking genauer von sonstigen Drittparteieinbettungen zu unterscheiden. Dies geht auf ein Klassifizierungsschema zurück, welches von F. Roesner im Jahr 2012 veröffentlicht wurde und in Abschnitt 7.1.1 näher betrachtet wird. Die Autoren erkennen[8], dass dieses Vorgehen aufgrund der Datenlage nicht sinnvoll ist, und berücksichtigen alle Formen der Drittparteieinbettungen als potenzielles Web-Tracking.

VERGLEICH MIT DEM LIVE WEB. Ähnlich wie in der vorliegenden Arbeit stellen die Autoren fest, dass ein nicht trivialer Teil an Drittparteianfragen (Third-Party Requests) im Archived Web fehlen, wenn diese mit dem Live Web verglichen werden.

TESTMENGE. Lerner et al. verwenden unterschiedliche Quellen zur Zusammensetzung der Testmengen für die verschiedenen Jahrgänge. So werden für die jeweiligen Jahre unterschiedliche Webseiten betrachtet, von denen eine gewisse Popularität in ihrer Zeit erwartet wird. Ein solches Vorgehen hat Vor- und Nachteile: Zwar kann auf diese Weise besser sichergestellt werden, dass tatsächlich nur populäre Webseiten betrachtet werden, auf der anderen Seite war ein Ziel dieser Arbeit, ein Entwicklungsbild des gesamten Internets zu zeigen und nicht aus-

8 „Thus, we learn that to study third-party web tracking in the past, due to missing data in the archive, we must consider all third-party requests, notonly those confirmed as trackers according to the taxonomy.“, Quelle: [103].

schließlich von populären Webseiten. Darüber hinaus ist zu berücksichtigen, dass solche Informationen nicht in der Quantität vorliegen, wie sie für diese Studie notwendig gewesen wäre.

KENNZAHLEN. Wie in der vorliegenden Arbeit wird von Lerner et al. ein Box-Whisker-Plot 6.10 für die Drittparteieinbettungen zur Verfügung gestellt. Die Ergebnisse Mittelwert, Median und Quartile stimmen mit den hier vorgestellten (Abbildung 6.1) überein.

ABDECKUNGSANALYSE. Bei einem Vergleich der Abdeckung durch Drittparteien erzielen die Studien vergleichbare Ergebnisse, wie in den Abbildungen 6.11 und 6.2 zu erkennen ist. Lener et al. geben[9] für die Top 20 der Drittparteien der Jahre 2000, 2005, 2008, 2010 und 2015 Abdeckungswerte von 12 %, 25 %, 45 %, 52 % und 71 % an. Zum Vergleich: In dieser Studie wurden in diesen Jahren Abdeckungswerte von 8,6 %, 20 %, 61 %, 75 % und 86 % gemessen.

RANGLISTE DER DRITTPARTEIEN. Lerner et al. geben eine grafische Übersicht über häufig eingebundene Drittparteien wie in Abbildung 6.12 zu sehen. In der vorliegenden Studie werden diese in Tabelle 6.3 zur Verfügung gestellt. Auch bei Lener et al. dominieren Doubleclick und Dienste von Google ab 2006 das Web. Die von Lerner et al. zusätzlich gefundenen Drittparteien (come.to, go.com, v3.com, allyes.com) zeigen Abdeckungswerte von unter 5 %, was weniger als 25 Webseiten entspricht, so dass diese nicht zu berücksichtigen sind.

GOOGLE-ANALYTICS Auch wenn das Verhältnis der Abdeckungswerte den hier gemessenen Daten entspricht, fallen diese deutlich geringer aus als erwartet. Dass für Google-Analytics für 2015 und 2016 nur Abdeckungswerte von unter 35 % angegeben wurden, widerspricht nicht nur den hier gemessenem Wert (2015) von 55.3 %, sondern auch den Ergebnissen von Englehardt und Narayanan [48], die bei einer Messung im Live Web auf 67 % gelangt sind. Übereinstimmend zeigt sich die Betrachtung des leichten Abwärtstrends von Google-Analytics ab dem Jahr 2011, wie dieser auch hier festgestellt wurde.

DOUBLECLICK So wie in Tabelle 6.3 eine vergleichsweise schwache Abdeckung von Doubleclick gezeigt wurde (25,2 %), gelangen auch Lerner et al. auf Werte zwischen 15 % und 18 %. Das Verhältnis von Google-Analytics zu Doubleclick (ca. 2:1) stimmt überein. In Abschnitt 6.3.2 wurde auf eine Messungenauigkeit bei Doubleclick hingewiesen. Ein Vergleich mit diesen Ergebnissen zeigt, dass dies in den archivierten Daten begründet liegt.

Ferner werden von Lerner et al. weitere Analysen präsentiert, die sich auf bestimmte Trackingmethoden konzentrieren, z.B. Analyse von Cookies oder

9 Aufgrund der fehlenden tabellarischen Auflistung wurden die Werte von Abbildung 6.11 abgelesen.

der Einsatz des localStore des Browsers (alternative Speichermethode). Diese Analysen sind vermutlich dem ersten Ansatz geschuldet, das Tracking in das oben erwähnte Klassifizierung einzuordnen, was anschließend verworfen wurde.

Das Fazit der Autoren deckt sich mit den Ergebnissen dieser Arbeit, die im Diskussionsabschnitt 6.4 erneut zusammengefasst werden:

> „[...] today's users browsing the web's popular sites encounter more trackers, with more complex behaviors, with wider coverage, and with more connections to other trackers, than at any point in the past 20 years."
> Quelle: [103, S. 15]

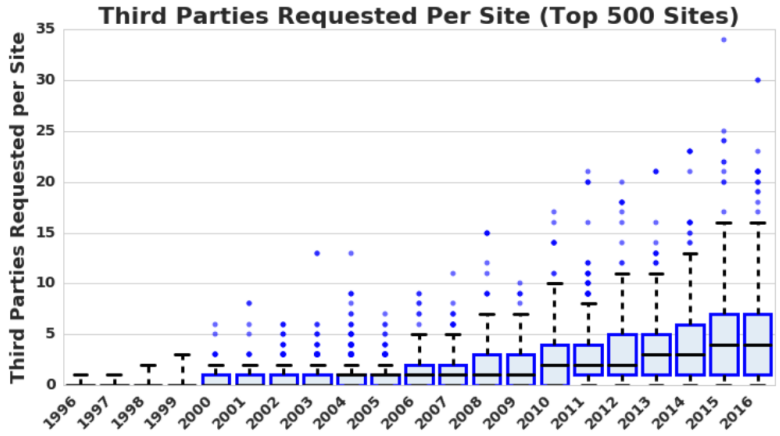

Abbildung 6.10: Box-Whisker-Plot zur Anzahl von Drittparteien entnommen aus [103, S. 12]

6.3.3 *Fazit zur Evaluation*

Ergebnis

Der Vergleich mit den Studien zeigt sowohl Gemeinsamkeiten und Unterschiede auf. Die Unterschiede prägen sich stärker aus, je weiter der Blick in die Vergangenheit geht.

Werkzeugdesign

In der ältesten vorliegenden quantitativen Studie von Krishnamurthy und Wills werden Tracker erwähnt, die während der Analyse nur schwach gemessen wurden. Grundsätzlich kommt hierfür das Werkzeug als Fehlergrund in Betracht. Aufgrund der Tatsache, dass diese Tracker auch von Lerner et al. nicht. gefunden wurden, schließt das Werkzeug als Fehlerquelle aus. Somit ergibt sich hier keine Verbesserungsmöglichkeit für Studien- oder Werkzeugdesign.

Inkonsistenzen untereinander

Der Vergleich der Studien ergab auch mögliche Inkonsistenzen untereinander. Dass von Chaabane et al. eine geringere Abdeckung durch Google gemessen wurde, als dies in der vorherigen Studie von Krishnamurthy und Wills und in der nachfolgenden Studie von Libert der Fall ist lässt an diesen Ergebnissen zweifeln. Die restlichen Ergebnisse sind schlüssig und stimmen mit den hier präsentierten Ergebnissen überein.

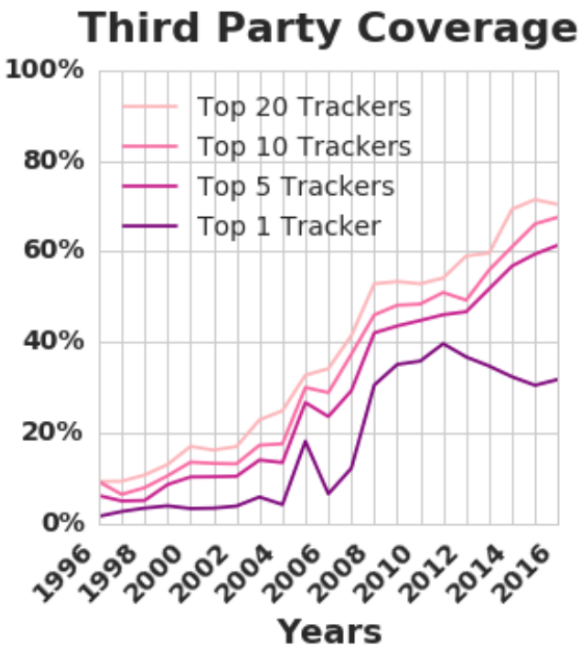

Abbildung 6.11: Abdeckungsanalyse zur Anzahl von Drittparteien entnommen aus [103, S. 13]

Übereinstimmende Ergebnisse

Bis auf wenige Ausnahmen entsprechen die Ergebnisse von Libert sowie Englehardt und Narayanan den retrospektiven Messungen. Ungenauigkeiten sind auf die Blockierung sowie auf Verweise nicht-archivierter Inhalte zurückzuführen. Eine grobe Abschätzung, wie stark die Differenz ist, ermöglicht ein Vergleich mit der Studie von Libert. Libert zeigt mit durchschnittlich 9,47 Einbettungen 51 % mehr als die hier gemessenen 6,27 pro Webseite. Zu berücksichtigen ist daher, dass die Ergebnisse eine Untergrenze für Web-Tracking angeben, und die tatsächlichen Werte oberhalb liegen können.

Fazit

Obwohl mit den Studien verschiedene Arten der Analysen durchgeführt wurden, zeigt der Vergleich mit Lerner et al. eine gute Übereinstimmung trotz abweichender Testmenge und komplett unterschiedlichem Werkzeugdesign. Die Evaluation erbrachte keine Hinweise darauf, mit einer abweichenden Methodik in Studie oder Werkzeugentwicklung bessere Ergebnisse zu erzielen.

6.4 DISKUSSION

Kennzahlen

Die ermittelten Kennzahlen in Tabelle 6.2 zeigen einen deutlichen Anstieg. In den Jahren 2005 bis 2015, in denen eine vergleichbare Anzahl betrachteter Webseiten vorliegen, haben sich die durchschnittlichen Einbettungen pro Drittpartei in dieser Zeit von 1,13 auf 6,61 erhöht.

Quartile

Insbesondere die Quartile als ein Maß der Verteilung sind von besonderem Interesse. Im Jahr 2005 waren Einbettungen auf der Webseite noch selten zu finden. Dies erkennt man an dem Wert 0 im ersten und zweiten Quartil

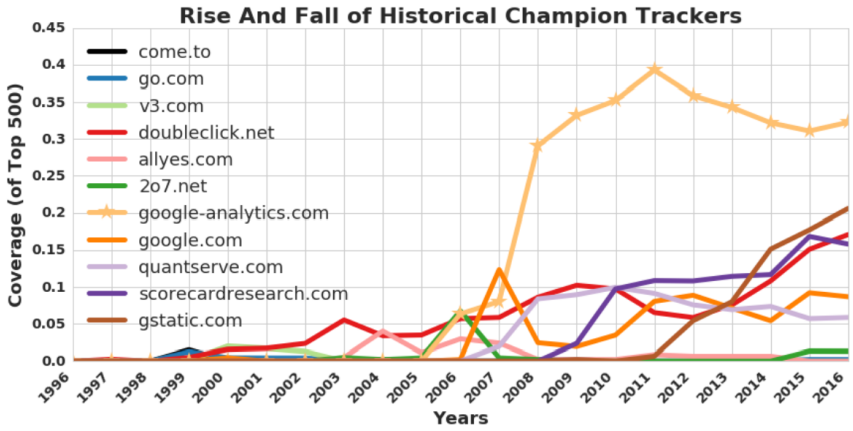

Abbildung 6.12: Übersicht häufig eingebundener Tracker bzw. Drittparteien entnommen aus [103, S. 13]

(Median). Demzufolge führten die Hälfte der vermessenen 10 318 Webseiten keine Einbettung durch. In den Jahren 2006 und 2007 startete, wie in Tabelle 6.3 zu erkennen ist, das starke Wachstum von Google-Analytics. Dies zeigt sich auch in den Kennzahlen: im Jahr 2006 war der Median bei 1 und das hintere Quartil stieg auf 2 und 2007 auf 3 an. Diese Bewegung setzt sich bis 2009 fort, in der die Ersteinbettung ins vordere Quartil rückt.

Ländervergleiche

Eine Betrachtung der durchschnittlichen Einbettungen der jeweiligen Top Level Domain eignet sich nur sehr eingeschränkt für einen Ländervergleich, wie in Abschnitt 6.2.6 bereits beschrieben wurde. Die Ergebnisse sind dahingehend schlüssig, da die Anzahl der Einbettung auf .com-Domains, die ursprünglich für kommerzielle Inhalte vorgesehen waren, deutlich über dem Durchschnitt liegen. So bewegt sich diese im .com-Fall im Bereich von 0,66 bis 7,1 und damit über dem Durchschnitt von 0,61 bis 6,61. Im Vergleich dazu zeigt die .org TLD, die ursprünglich für nicht-kommerzielle Inhalte vorgesehen war, eine unterdurchschnittliche Einbettungsrate von 0,24 bis 5,24.

Zuwachs der Einbettungen

Während der Zuwachs an Drittparteieinbettungen deutlich wird ist zu hinterfragen, welche Drittparteien am häufigsten eingebunden wurden. Es zeigt sich, dass bei den 9783 Webseiten, die 2004 analysiert wurden, die am häufigsten eingebettete Drittpartei (doubleclick.net) 306-mal im Webauftritt integriert wurde. Dies entspricht einer Abdeckung von nur 5 % und fällt damit vergleichsweise gering aus. Auf dem zweiten Platz befindet sich googlesyndication.com, welches 259-mal eingebunden wurde und auf Platz 3 (rambler.ru) findet sich auf 246 Webseiten.

Rückgang

2006 zeigte eine deutliche Änderung: Der Dienst Google-Analytics, der 2004 noch nicht existierte und 2005 nur 13-mal gesichtet wurde, fand sich schon 2006 auf 1404 Webseiten. Diese Änderung bildet den Auftakt eines starken Anstiegs des Google-Analytics-Dienstes in den darauffolgenden Jahren 2006 bis 2015. Ein leichter Rückgang in den Jahren 2014 und 2015 ist mit einer Verlagerung auf andere Google-Dienste zu erklären.

Abdeckung der Tracker

Die Abdeckungsanalyse (Abbildung 6.2) zeigt, wie in den Jahren 2010 bis 2015 allein der erste Platz (Google-Analytics) hohe Abdeckungswerte erreicht. Die Abdeckung durch die verbleibenden 49 Plätze stieg im Jahr 2010 um ca. 20 % und im Jahr 2015 um ca. 32 %. Bei dem Vergleich der Netzwerkgrafiken vom Jahr 2005 und 2007 (Abb. 6.4 und Abb. 6.5) zeigt sich ein deutlicher Zuwachs von Google-Analytics (K) der zum Jahr 2010 (Abbildung 6.6) weiter zunimmt.

Funktionale Bestandteile

Ebenfalls beachtlich sind die Abdeckungswerte von Diensten, die nicht direkt Web-Tracking zugeordnet werden. Ein Beispiel ist die Bereitstellung von Schriftarten durch den Google-Dienst fonts.googleapis.com, der im Jahr 2015 bei 2023 Webseiten gesichtet wurde.

Forschungsfrage

Abschließend wird die in Abschnitt 4.2 formulierte Forschungsfrage RQ-1 beantwortet:

> **In welcher Weise hat sich Web-Tracking in den vergangenen Jahren ausgebreitet?**

Evaluation

Neben den bestehenden quantitativen Analysen (Krishnamurthy und Wills, Chaabane et al., Libert, Engelhardt und Narayanan) können archivierte Webseiten retrospektiv bzgl. des Einsatzes von Web-Tracking ausgewertet werden. Dabei werden spezielle Anforderungen an das Analysewerkzeug gestellt.

Einschränkungen

Es zeigte sich, dass eine qualitative Analysen der Webseiten aufgrund durch die Archivierung bedingten Einschränkungen nicht vollständig durchführbar ist. Die Beantwortung der Forschungsfrage nach der Ausbreitung von Web-Tracking lässt sich infolgedessen nur eingeschränkt beantworten.

Ergebnis

Bedingt durch diese Einschränkungen wurde sich auf den messbaren Teil konzentriert: die Einbettung von Drittparteien. Die Messung zeigt eine mindestens sechsfache Zunahme der Drittparteien zwischen den Jahren 2005 bis 2015. Diese bedeutet zwar zunächst keine Zunahme von Web-Tracking, allerdings ist durch die Art der Drittparteien bzw. durch die Monopolisierung der Zweck der Einbettung ableitbar. Alleine Google-Analytics zeigt eine Zunahme von 0,1 % bis 55 % in der Zeitspanne von 2005 bis 2015. Damit spiegelt sich die Präsenz und Bedeutung von Web-Tracking bzw. Web-Analytics wider.

		'00	'01	'02	'03	'04	'05	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15
A	doubleclick.net	276	345	339	287	306	346	406	643	851	1868	1738	1567	1223	1890	2978	2707
B	rambler.ru	113	152	184	215	246	264	296	317	312	312	296	286	254	213	178	157
C	hitbox.com	81	93	88	111	95	134	152	154	119	63	30	16	4	1		
D	akamai.net	65	160	149	142	130	143	154	135	103	78	72	69	66	76	50	40
E	imgis.com	61	18														
	bfast.com	57	68	41	34	24	17	9									
F	spylog.com	52	148	135	127	125	113	127	113	84	88	62	51	38	34	23	16
	extreme-dm.com	47	59	63	81	95	106	113	78	67	52	36	31	32	25	17	13
	flycast.com	46	36														
	linkexchange.com	46	37	25	23	6	1	1	1	1							
G	list.ru	45	110	150	174	192	195	225	227	199	162	133	106	90	70	51	42
	register.com	43	66	46	36	29	21	5	2	3							
	thecounter.com	42	57	21	18	18	12	2	2	2	1						
	akamaitech.net	41	32	15	14	8	7	7	7	6	3	4	1				
	linksynergy.com	29	41	32	32	27	35	32	27	23	14	13	12	10	10	8	2
	webtrendslive.com	12	72	46	40	54	77	89	102	106	105	96	116	114	93	70	51
	216.21.232.20	15	60	40	32	31	17										
	209.10.130.68	15	57														
I	imrworldwide.com	28	55	68	97	149	166	199	222	265	266	285	428	424	380	293	240
	qksrv.net		41	72	74	53	13	7	4	1	2	2	1	1			
	hotlog.ru			67	102	129	110	101	80	65	41	34	31	25	21	16	12
	fastclick.net	2	10	59	57	70	63	61	66	58	56	61	45	43	35	20	16
	advertising.com	11	14	54	59	85	62	65	102	121	108	120	108	98	80	55	51
	nedstatbasic.net	17	30	53	73	65	53	16	8	6	3	3	1	1	1		
	oingo.com		6	49	76	74	41	24									
	facetz.net			47	70	96	82	68	50	45	26	20	17	18	14	10	9
H	googlesyndication.com				32	259	523	813	995	977	946	924	970	902	858	865	966
L	207.net			15	24	84	152	225	318	351	337	414	444	406	378	294	224
	yadro.ru			12	43	72	119	200	255	288	311	331	349	347	345	331	314
	falkag.net				2	35	101	132	46	12							
M	google.com	5	10	15	26	38	96	161	1587	466	604	736	1241	2115	2212	2397	2231
K	google-analytics.com						13	1404	3157	4567	5394	6072	6666	7040	6881	6087	5940
	statcounter.com				3	14	69	132	125	122	111	91	81	77	75	64	64
	gemius.pl		4	14	37	56	67	106	172	177	175	260	343	359	375	365	336
	atdmt.com		10	11	31	47	68	108	141	159	149	163	175	136	119	72	40
	scanalert.com				4	22	40	66	128	192	188	176	141	125	105	70	54
N	quantserve.com								90	376	689	793	851	861	816	687	589
	googleadservices.com							13	38	136	321	643	1058	1306	1530	1572	1392
	tacoda.net						2	66	82	133	106	89	72	37	15	17	10
P	ajax.googleapis.com									20	222	623	1199	1711	2076	2170	2153
	addthis.com			1					20	115	220	317	369	432	451	474	446
	revsci.net						4	48	48	120	167	208	235	244	222	172	189
O	facebook.com										67	580	1638	1826	1604	1403	1083
	scorecardresearch.com										86	448	777	988	1124	1196	1190
Q	facebook.net											161	851	1611	1952	2137	2400
	fbcdn.net									1	52	121	777	497	1136	655	251
	twitter.com										34	137	636	1116	1440	1598	1547
	googletagservices.com												17	317	836	1353	1681
	fonts.googleapis.com											5	61	308	773	1464	2023
	googleusercontent.com											2	41	182	537	522	29
	gstatic.com									1	11	33	88	262	357	1081	1239
	googletagmanager.com														327	979	1782
	akamaihd.net												6	17	90	435	715

Tabelle 6.3: Top 15 der häufigsten eingebetteten Drittparteien, die Top 5 sind grau hinterlegt.

Jahr	Gesamt		com		net		ru		org		de	
	#	$\bar{x}$	#	$\bar{x}$	#	$\bar{x}$	#	$\bar{x}$	#	$\bar{x}$	#	$\bar{x}$
2000	6504	0,61	3506	0,66	174	1,26	171	2,74	221	0,24	274	0,45
2001	7741	0,77	4192	0,81	238	1,33	254	2,96	252	0,43	299	0,68
2002	8314	0,81	4479	0,83	284	1,36	287	3,17	277	0,39	303	0,72
2003	8923	0,85	4816	0,84	320	1,31	315	3,34	288	0,43	309	0,85
2004	9783	0,99	5339	0,99	362	1,48	337	3,44	309	0,48	325	1,07
2005	10318	1,13	5603	1,16	377	1,79	369	3,36	337	0,62	339	1,16
2006	10817	1,45	5800	1,54	405	2,1	398	3,91	352	0,87	351	1,52
2007	11014	1,92	5937	2,02	412	2,59	416	4,22	357	1,53	355	1,98
2008	11352	2,08	6249	2,21	434	2,6	410	4,4	369	1,46	345	2,09
2009	10899	2,53	5950	2,75	418	3,02	405	4,76	355	1,97	342	2,3
2010	10964	3,07	5936	3,39	415	3,49	403	5,14	354	2,23	345	2,87
2011	11208	3,91	6119	4,33	422	4,37	411	6,11	364	3,05	348	3,49
2012	11174	4,63	6064	5,13	427	4,7	413	6,67	369	3,39	352	4,02
2013	11174	5,44	6105	5,94	420	5,51	413	7,5	361	3,98	353	5,07
2014	10861	6,27	5940	6,73	417	6,54	408	8,38	353	4,76	343	6,2
2015	10736	6,61	5889	7,1	410	6,56	410	8,46	353	5,24	342	6,58

Tabelle 6.4: Betrachtung der fünf häufigst vertretenen Top Level Domains.

Live Web	Archived Web
adition.com	adition.com
mxcdn.net	mxcdn.net
ioam.de	ioam.de
google-analytics.com	google-analytics.com
soundcloud.com	soundcloud.com
meetrics.net	d1z2jf7jlzjs58.cloudfront.net
googletagservices.com	
yieldlab.net	
doubleclick.net	
mediaplex.com	
westlotto.com	
googlesyndication.com	
2mdn.net	
adsafeprotected.com	
dotomi.com	
parsely.com	
sndcdn.com	
criteo.com	

Tabelle 6.5: Vergleich der Drittparteien zwischen Live Web zu Archived Web.

Teil II

WEB-TRACKING IN DER GEGENWART

Im Folgenden wird ein Gesamtüberblick zu Verfahren und Schutzmaßnahmen des Web-Trackings geboten. Anschließend werden die Ergebnisse von zwei quantitativen Studien präsentiert: In Studie A wird eine Prüfung von Datenschutzerklärungen auf Hochschulwebseiten durchgeführt und in Studie B werden Webseiten aus dem Gesundheitswesen näher betrachtet.

7 VERFAHREN UND SCHUTZ

Zusammenfassung: In diesem Kapitel werden aktuelle Web-Tracking-Verfahren und verfügbare Schutzmaßnahmen näher betrachtet. Ziel ist es, durch die Zusammenführung verschiedener Quellen einen umfassenden Überblick zu bieten und erweiterte Nachforschungen zu ermöglichen. Nach einer Einleitung, mit allgemeinen Informationen zu Web-Tracking, werden zustandsbasierte und zustandslose Tracking-Verfahren betrachtet. Anschließend wird sich detailliert mit verschiedenen Schutzmethoden auseinandergesetzt.

7.1 EINLEITUNG

Web-Bugs

Martin et al. [114] analysierten 2001[1], inwieweit Einbettungen[2] auf technischer Ebene zu einer Weitergabe von Informationen über den Webseitenbesucher führt. Im Zuge der Publikation ist ebenfalls ein Werkzeug zur Detektion solcher Einbettungen [7] entstanden, welches als Schutzwerkzeug einzuordnen ist. Dieses frühe Beispiel zeigt, wie Tracking als potentielle Gefahr für die Privatheit der Nutzer erkannt und gleichzeitig eine mögliche Gegenmaßnahme aufgezeigt wurde.

Entwicklung

Genau in dieser Weise hat sich im Laufe der Zeit ein akademischer Wettstreit ergeben, um neue Arten des Trackings zu finden sowie Schutzmöglichkeiten zu entwickeln. Die Weiterentwicklung der Web-Technologien und damit begleitende Funktionserweiterungen der Browser begünstigte stets neue Arten und Formen von Tracking-Verfahren.

Toolkits

Es ist zu bezweifeln, dass alle diese „Schwachstellen“ einen Weg in die kommerzielle Trackingwelt gefunden haben. Allerdings demonstrieren und analysieren Nikiforakis et al. [133] fertige Toolkits, die von Werbefirmen direkt eingesetzt werden können. So müssen nur diese Toolkits einmalig erweitert werden, um auf einen Schlag die Qualität mehrerer Trackingdienste zu verbessern. Aus diesem Grund ergibt sich aus jeder Trackingmöglichkeit auch eine Gefahr unabhängig von der Wahrscheinlichkeit der Umsetzung.

7.1.1 *Klassifikation von Web-Tracking-Verfahren*

Klassifikation von Roesner et al.

Roesner et al. [153] führten 2012 eine Klassifizierung von Web-Tracking-Verfahren ein. Dies sollte die Lücke schließen, dass bis zu diesem Zeitpunkt eine techniknahe Unterscheidung der verschiedenen Verfahren fehlte. Es

1 Nach [7] ist die Arbeit schon im Jahr 2001 entstanden, wurde jedoch erst 2003 veröffentlicht.
2 Martin et al. bezeichnen diese als „Web-Bugs“.

wurde eine Erkennungssoftware implementiert, welche die Verbreitung dieser Klassen im Web misst. Infolgedessen wurde neben einer theoretischen Arbeit auch eine quantitative Studie durchgeführt, um das Unterscheidungssystem zu evaluieren.

Klassen von Trackern

Im Folgenden werden die Klassen von Trackern beschrieben. Genauere Informationen sowie Abbildungen zu den Verfahren können von Roesner et al. [153] entnommen werden. Es wurden fünf Klassen von Tracking-Verfahren unterschieden:

(A) ANALYTICS. Eine Analysesoftware als Teil des Webauftritts, welche zur Messung von Besucherzahlen eingesetzt wird. Während es sich bei den anderen Klassen typischerweise um domänenübergreifende Verfahren (Cross-Site) handelt, wird hier nur der eigene Webauftritt überwacht (Within-Site).

(B) VANILLA. Einbettung einer Drittpartei auf der Webseite, die üblicherweise durch eine Speicherung eine domänenübergreifende Wiedererkennung realisiert.

(C) FORCED. Durch den direkten Aufruf einer Webseite (z. B. in einem Popup) wird die Datenweitergabe an einen Dritten realisiert.

(D) REFERRED. Diese Klasse ist keine eigene Trackingform, sondern beschreibt eine spezielle Form der Informationsweitergabe auf Basis eines Trackers der Klassen B, C oder E.

(E) PERSONAL. Hierbei handelt es sich um teils funktionale Einbettungen in Webseiten wie z. B. für Social Widgets (Like-Button). Es unterscheidet sich von B dadurch, dass es nicht vordergründig zum Tracking genutzt wird, aber ein vergleichbares Potential dafür bietet.

Tracker können mehrere Klassen umfassen. In einer quantitativen Analyse wurden 14 verschiedene Kombinationen der Klassen (A, A+B, A+B+D, A+E, etc.) gemessen.

Unschärfe

Die Zugehörigkeit lässt sich nicht stets eindeutig feststellen. Die Autoren nennen hierfür das Beispiel `quantserve.com`, welches den Webseitenbetreibern als Analysetool für ihre eigene Seite dient (Klasse A). Dies kann auch als ein Cross-Site-Tracker eingesetzt und integriert werden (Klasse B). Unter welcher Kategorie dieser Tracker eingeordnet wird, ist von der Einsatzweise abhängig. Diese Unschärfe begründet vermutlich, dass diese Klassifizierung nur wenig in der Literatur berücksichtigt wird.

7.1.2 *Gründe für Web-Tracking*

Gründe

Wie in Abschnitt 1.2 festgehalten wurde, ist eine nähere Betrachtung der Gründe für Web-Tracking nicht Bestandteil dieser Arbeit. Bujlow et al. [28, S. 15] geben eine Übersicht der Gründe – wesentlich darunter sind:

ONLINE WERBUNG. Die Daten der Tracker dienen zur Bestimmung von passenden Werbeeinblendungen auf Webseiten. Weiteres dazu erörtern Bilenko et al. [21].

WEB ANALYTICS. Nutzungsdaten können dazu verwendet werden, das Angebot sowie den gesamten Webauftritt zu optimieren. Von Interesse ist dabei, woher ein Besucher kommt, welche Inhalte näher bzw. länger betrachtet wurden und was dabei besonders fokussiert wurde.

USABILITY TESTS. Mittels so genannter A/B-Tests kann die Bedienungsfreundlichkeit der Webseite ermittelt und verbessert werden. Dabei wird zufällig ausgewählten Besuchern eine abweichende Variante des Webauftritts ausgeliefert und geprüft, ob sich Vorteile verglichen zur ursprünglichen Version zeigen.

BEWERTUNG DER FINANZKRAFT. Im Jahr 2012 wurde bekannt[3], dass die Schufa, eine privatwirtschaftliche deutsche Auskunftei, Informationen auf Facebook und Twitter in die Beurteilung der Kreditwürdigkeit einer Person einbeziehen. Der Kreditgeber Kreditech ermittelt[4] die Wahrscheinlichkeit des Kreditausfalls basierend auf Browserinformationen.

7.1.3 *Anforderungen an Trackingverfahren*

Anforderungen an Tracker

Nach Schneider et al. [161] müssten Web-Tracking-Verfahren die Anforderungen Reichweite, Eindeutigkeit, Stabilität und Robustheit erfüllen.

REICHWEITE. Damit ist gemeint, dass die Verfahren auf möglichst vielen Plattformen funktionieren und nicht nur auf kleine Nutzergruppen festgelegt sind. Verfahren, die nur in Browsern funktionieren, die einen verhältnismäßig kleinen Nutzerkreis haben, würden dieser Anforderung demnach nicht genügen.

EINDEUTIGKEIT. Das Verfahren muss zwischen den getrackten Zielen eindeutig unterscheiden können. Diese Anforderung ist insbesondere in Hinblick auf Fingerprinting-Verfahren von Bedeutung.

STABILITÄT. Der Zeitraum, in dem das Tracking zuverlässig funktioniert, soll möglichst groß sein.

ROBUSTHEIT. Das Tracking-Verfahren soll möglichst resistent gegen zufällige Änderungen am System oder durch den Benutzer sein. Ein Beispiel ist die Robustheit gegen das aktive Löschen von Cookies auf dem System durch den Nutzer.

Weitere Anforderung

Diese Auflistung muss um die Anforderung der „Sichtbarkeit“ erweitert werden. Sofern ein unbemerktes Tracking durchgeführt und der Tracker nicht erkannt werden soll, wird eine möglichst geringe Sichtbarkeit vom Tracker angestrebt. Grund für diese Erweiterung ist, dass insbesondere speicherbasierte Verfahren über eine höhere Sichtbarkeit verfügen als Fingerprinting-Verfahren. Während auf Speicherung basierende Tracking-

3 http://www.faz.net/aktuell/wirtschaft/pruefung-der-kreditwuerdigkeit-schufa-will-facebook-profile-auswerten-11776537.html, abgerufen am 27.01.2018.

4 https://www.welt.de/finanzen/verbraucher/article139671014/Gegen-Kreditech-ist-die-Schufa-ein-Schuljunge.html, abgerufen am 27.01.2018.

Verfahren stets durch eine Änderung am System auffällig werden können, sind Fingerprinting-Verfahren, je nach Art, nur schwer feststellbar.

7.2 Übersicht zu Web-Tracking

Überblick zu Verfahren

Web-Tracking Verfahren unterteilen sich in die zustandsbasierte („Supercookies“) und zustandslose („Fingerprinting“) Verfahren. Diese werden in den Abschnitten 7.3 und 7.4 näher betrachtet. In Abbildung 7.1 wird eine Kategorisierung verschiedener Veröffentlichungen vorgenommen und zu einem Gesamtbild zusammengesetzt. Strategien zum Schutz vor Web-Tracking werden ab Abschnitt 7.5 beschrieben.

7.3 Zustandsbasiert – Supercookies

Supercookies

Wie Bujlow et al. [28] beschreiben, bedurften die ersten bekannt gewordenen Trackingverfahren einer „Markierung“ auf dem zu trackenden System. Gemeint ist damit eine möglichst persistente Änderung (z.B. Speicherung), die eine Wiedererkennung über eine längere Zeitspanne ermöglicht. Hierfür wird der Begriff „Stateful“ u. a. in Mayer und Mitchell [115] verwendet und beschreibt damit alle Arten von Tracking-Verfahren, die auf einer Zustandsänderung des zu trackenden Systems (client-state) beruhen. Von Pan et al. [139] werden diese als „stateful third-party web tracking“ bezeichnet. Gleichbedeutend findet sich auch der Begriff „Evercookie“ in der Literatur, welcher durch einen Demonstrator [93] von Samy Kamkar geprägt wurde.

Typen

Eine genauere Unterscheidung zwischen den verschiedenen Stateful-Verfahren wird von Bujlow et al. durchgeführt, die in sitzungs-, speicher- und pufferbasierte Verfahren differenzieren. Grundsätzlich findet bei allen eine Systemänderung statt; die Übergänge sind dabei fließend und können sich zwischen den verschiedenen User-Agents unterscheiden.

7.3.1 *Sitzungsbasiert*

Erzeugung

Die Herstellung einer Sitzung (in Bezug auf HTTP), also der Erhalt eines Kontextes über mehrere Webseitenaufrufe hinweg, ist an vielen Stellen technisch gewollt und erwünscht: Die Bestellung in einem Internetshop umfasst üblicherweise mehrere Seitenaufrufe bis zum vollständigen Abschluss. Die gleichen Mechanismen, die dem Erhalt eines Warenkorbs bis zum Bestellabschluss gewährleisten, können zur Verfolgung und Überwachung des Besuchers angewendet werden.

Sitzungsschlüssel

Ein Umsetzungsbeispiel ist die Nutzung eines eindeutigen Sitzungsschlüssels, welcher an jeden dynamisch generierten Link auf der Webseite angehangen wird. Vor der Einführung von Cookies im Jahr 1994 war dies der einzige verfügbare Weg eine solche Sitzung zu etablieren. Modernere Beispiele neben dem oben beschriebenen sind auch Zugangsdaten (web-form authen-

tication) oder das `windows.name` DOM-Property; beide wurden von Bujlow et al. [28] näher beschrieben.

Daten einer Sitzung werden als flüchtig betrachtet: Informationen, die nach Schließung des Browsers durch den Benutzer verloren gehen würden, sind aus diesem Grund für ein längerfristiges Tracking ungeeignet. Sie widersprechen somit der Stabilitätsanforderung aus Abschnitt 7.1.3.

7.3.2 *Speicherbasiert*

Persistenz

Diese Verfahren basieren auf persistenten Änderungen am System des Nutzers. Durch dauerhafte Veränderung der Daten auf dem Speichermedium wird eine langfristige Wiedererkennung durch den Tracker ermöglicht.

HTTP-Cookie

Die älteste Form der clientseitigen Speicherung von Informationen ist das HTTP-Cookie, dessen Entstehungsgeschichte bereits in Abschnitt 3.2 beschrieben wurde. Nach der Speicherung eines HTTP-Cookies mit frei wählbarem Inhalt, übermittelt der Browser diesen in allen folgenden Anfragen an die jeweilige Domain. Die Art, wie Cookies auf dem System gespeichert bzw. verwaltet werden, ist allein von den Designentscheidungen der Browserentwickler abhängig. Nach RFC 6265 wird empfohlen, eine Möglichkeit zur Verwaltung bzw. zur Löschung zu schaffen: „User agents SHOULD provide users with a mechanism for managing the cookies stored in the cookie store.“, Quelle: [12].

Stabilität und Robustheit

Eine einfache Löschung solcher Identifikationsmerkmale ist aus offensichtlichen Gründen nicht im Sinne der Trackingunternehmen und widerspricht der Stabilitäts- und Robustheitsanforderung an ein Trackingverfahren (vgl. Abschnitt 7.1.3). Aus Perspektive der Tracker muss sichergestellt sein, dass auch nach möglichen Bereinigungsvorgängen eine Wiedererkennung durchgeführt werden kann. Der Einsatz unterschiedlicher Speichertechnologien wurde auch wissenschaftlich behandelt.

FLASH COOKIES. Im Jahr 2011 wurde von McDonald und Cranor [117] genauer analysiert, wie das Adobe Flash Plugin zur langfristigen Speicherung von Informationen und so auch zum Tracking verwendet werden kann. Diese sind, anders als HTTP-Cookies, für ein Cross-Browser-Tracking geeignet, was bedeutet, dass auch bei einem Wechsel des Browsers (z. B. vom Microsoft Internet Explorer zum Mozilla Firefox) eine Wiedererkennung des Nutzers möglich ist.

JNLP PERSISTENCESERVICE. Als Bestandteil der Java-Laufzeitumgebung[5] ermöglichen die JNLP-Dienste eine Speicherung von Informationen auf dem ausführenden System, die ebenfalls zum Tracking verwendet werden können.

HTML5. Die Entwicklung von HTML5 ermöglicht neue Interaktionsmöglichkeiten des Browsers mit dem umliegenden System. Dies umfasst

5 https://docs.oracle.com/javase/8/docs/jre/api/javaws/jnlp/javax/jnlp/PersistenceService.html, zuletzt abgerufen am 02.01.2018.

weitere Speichermöglichkeiten, deren Nutzung zum Tracking von Ayenson et al. [9] näher betrachtet werden.

USERDATA STORAGE. Bei dieser Methode handelt es sich um eine proprietäre Speichertechnik, die im Internet Explorer Einsatz findet bzw. gefunden hat, jedoch seit der Version 7 nicht länger zum Einsatz kommt [28].

7.3.3 *Pufferbasiert*

Zwischenspeicher

Zur Entlastung des Servers, zur Minimierung des Netzwerkaufkommens und zur Verbesserung der Geschwindigkeit werden bereits geladene Ressourcen auf dem System zwischengespeichert (HTTP Caching). Jedoch zeigt Felten et al. im Jahr 2000 [53] wie genau dieses Caching als invasiver Cookie-Ersatz dient. Diese Verfahren nutzen Pufferungseffekte von Browser und Betriebssystem aus, die zu einer Verhaltensänderung des Browsers führen. Bujlow et al. [28] unterscheiden dabei zwischen einem Web, DNS und einem Operational Cache.

Web cache

Historie

Besuchte Webseiten werden vom Browser in einer Historie gespeichert, die dazu dient, besuchte Links farblich zu markieren. Janc et al [88] zeigen, wie dieses Vorgehen ausgenutzt werden kann, um die Browserhistorie eines Besuchers auszulesen. Nach der Reaktion der Browserhersteller diese Schwachstelle zu beseitigen, präsentierten Weinberg et al. einen weiteren Seitenkanalangriff [197].

Dynamische Inhalte

Ebenso ist es als Webseitenbetreiber möglich, zufällige Bilder oder JavaScript-Dateien mit eindeutigen Identifikationsmerkmalen (IDs) zu versehen und auszuliefern. Beim erneuten Besuch der Webseite verwendet der Browser diese gepufferten Inhalte. Je nach Umsetzung wird somit ein gezielter Aufruf ausgelöst oder das Ausbleiben eines entsprechenden enthüllt die gepufferten Informationen.

Protokolle

Die HTTP-Felder `Last-Modified` und `ETag` [57] dienen der Umsetzung von Pufferstrategien. So wird eine Ressource nur bei Bedarf neu geladen bzw. sofern sich dessen Inhalt verändert hat. Um diese Änderung festzustellen, übermittelt der Browser über die zwei genannten Felder die Version im Pufferspeicher. Der Webserver kann nun entscheiden, ob ein erneutes Zusenden notwendig ist. Allerdings lassen sich diese Felder auch für Identifikationsmerkmale nutzen, da der Betreiber dessen Inhalte frei wählen kann und somit der Verhaltensweise eines klassischen Cookies entspricht.

DNS Cache

Systemspeicher

Bujlow et al. [28] verweisen bzgl. des DNS Cache nur auf zeitbasierte Angriffsformen. Dabei wird die Zeit einer DNS-Auflösung gemessen und so

geprüft, ob zuvor eine Auflösung stattgefunden hat. In Abschnitt 8.2.5 wird ein weiterer Angriff beschrieben, der innerhalb der Literatur bislang nicht näher betrachtet wurde.

Operational cache

Funktionale Speicherungen

Dabei handelt es sich um eine Speicherung, die nicht Teil der Inhaltsdaten im Web sind, sondern aus funktionalen Gründen gespeichert werden.

REDIRECT. Weiterleitungen werden durch den Webserver vom Browser protokolliert und vermerkt. Wird die ursprünglich gewählte Webseite zu einem späteren Zeitpunkt erneut angewählt, verwendet der Browser das zuvor gespeicherte Weiterleitungsziel [121, S. 61]. Durch die Verknüpfung des Weiterleitungsziels mit einer zufällig gewählten ID ermöglicht die Übermittlung beim zweiten Mal die Identifikation.

HTTP AUTHENTICATION. Die Authentifizierung (Basic Authentication) über HTTP, spezifiziert in RFC 7236 [150] bzw. RFC 2617 [129], erfolgt durch die Übermittlung einer Base64 kodierte Zeichenfolge die Benutzername und Passwort enthält. Einmal eingegeben, wird diese für eine festgelegte Zeitspanne bei jedem Webseitenaufruf erneut übermittelt. Diese Authentifizierungsdaten werden separat vom Browser gespeichert. Mithilfe von JavaScript werden Daten in diesen Speicher abgelegt, ohne das übliche Eingabefenster erscheinen zu lassen[6].

HSTS. Ist eine Webseite nur über HTTP und nicht über HTTPS abrufbar, kann der Browser nicht prüfen, ob dies stets der Fall ist oder ob diese einem SSLStrip-Angriff[7] unterliegt. Findet ein solcher Angriff statt, wird dem Browser signalisiert, dass die Webseite nur unverschlüsselt erreichbar ist. Der Einsatz von HTTP Strict Transport Security [78] bewirkt, dass bei erstmaligem Abruf einer Webseite von einem Webserver der Browser die Information speichert, ob die Domain mittels SSL/TLS abrufbar ist. Wird die Webseite zu einem späteren Zeitpunkt erneut geöffnet, wird vom Browser auf eine verschlüsselte Verbindung bestanden und die unverschlüsselte Version abgelehnt. Es zeigt[8] sich, dass in diesen HSTS Zustandsspeicher Daten für Web-Tracking genutzt werden können.

TLS WIEDERAUFNAHME. Eine Session-ID im SSL/TLS-Protokoll kann genutzt werden, Nutzer bei ihren Besuch über verschiedene Domains zu verfolgen[9]. Nach RFC 5246 [151, S. 93] kann eine solche Session-ID bis zu 24 Stunden aktiv sein.

6 http://blog.jeremiahgrossman.com/2007/04/tracking-users-without-cookies.html, abgerufen am 28.01.2018.

7 http://www.linux-community.de/Internal/Nachrichten/Sslstrip-taeuscht-HTTPS-Verbindung-vor, abgerufen am 28.01.2018.

8 http://www.leviathansecurity.com/blog/the-double-edged-sword-of-hsts-persistence-and-privacy, abgerufen am 28.01.2018.

9 https://trac.torproject.org/projects/tor/ticket/4099, abgerufen am 28.01.2018.

7.3.4 *Hilfsmittel*

Sync & Respawn

Sofern Trackingunternehmen sich gegenseitig in der Identifikation von Nutzern unterstützen, wird dies als Cookie Syncing bezeichnet. Die Wiederherstellung von Cookies auf Basis anderer gespeicherter Daten (z. B. LSO-Cookies) heißt Cookie Respawning.

Cookie-Syncing

Cookie-Syncing

Acar et al. [2] beschreiben Cookie-Syncing und messen dessen Einsatz durch eine Analyse von 100 000 Webseiten. Bei Cookie-Syncing (bzw. Synchronisation) handelt es sich um einen Austausch eindeutiger Identifikatoren von Nutzern zwischen Trackern. Dabei übermittelt eine Domain A nach einer erfolgreicher Identifikation des Besuchers ein entsprechendes Identifikationsmerkmal (z. B. eine ID) an Domain B. Dieser manuelle Austausch ist notwendig, weil die Same-Origin Policy keinen direkten Zugriff von Domain B an Cookies der Domain A zulässt. Von Google wird dieser Vorgang Cookie Matching[10] genannt.

Cookie-Respawning

Cookie-Respawning

Das Respawning (englisch: „neu starten“) von Cookies wurde erstmalig von Soltani et al. [168] beobachtet. Dabei stellen mehr als 50 % der analysierten Webseiten ein gelöschtes HTTP-Cookie durch ein Flash-Cookie wieder her. Es ist anzumerken, dass zum Zeitpunkt der Studie (2009) Adobe Flash ein häufig installiertes Plugin war. Im Jahr 2011 wurden weitere Speichertechniken bzgl. dieses Verhaltens analysiert [9]. Auch Acar et al. [2] messen u. a. Cookie-Respawning im Web.

7.4 ZUSTANDSLOS – FINGERPRINTING

Fingerprint

Der Fingerabdruck des Menschen ist ein biometrisches Merkmal, welches nach aktuellem Wissen eine eindeutige[11] Identifizierung ermöglicht. Im Jahr 2010 zeigte Eckersley [44], dass unter 500 000 Browsern 83,6 % der Nutzer eindeutig identifiziert werden konnten ohne eine Speicherung auf dem System des Nutzers durchzuführen. Dies wurde durch aktives Fingerprinting umgesetzt, indem er Browser auf vielfältige Weise ausgelesen und zur Preisgabe von Konfigurationen und Eigenschaften provoziert wurde. Beispiele sind die installierten Browserplugins, verfügbaren Schriftarten, gesendeten HTTP-Header, etc. Diese werden zur Generierung eines eindeutigen Fingerabdrucks verwendet. Einen Überblick gibt Abbildung 7.2.

Abgrenzung

Findet keine Provokation zur Preisgabe von eindeutigen Merkmalen statt und der Besucher wird lediglich anhand der Informationen identifiziert, die

10 https://developers.google.com/ad-exchange/rtb/cookie-guide, abgerufen am 18.01.2018.

11 Nur in wenigen dokumentierten Fällen weisen Menschen schon von Geburt an keinen Fingerabdruck auf.

freiwillig übermittelt werden, wird möglicherweise ein passives Fingerprinting angewendet.

7.4.1 *Aktive FP-Verfahren*

Informationsquellen

Mayer und Mitchell [115] listen Informationsquellen auf, die für aktives Fingerprinting genutzt werden können:

- Betriebssystem,
- CPU Typ,
- User-Agent,
- Zeitzone,
- Anzeigeeinstellungen,
- installierte Schriftarten,
- installierte/aktivierte Plugins,
- unterstützte MIME-Typen,
- Cookiebehandlung.

Bezugsmöglichkeiten

Diese Informationen können auf unterschiedliche Weise ausgelesen werden. Die Übermittlung des User-Agent ist beispielsweise Teil des HTTP-Headers. Der Agent kann wiederum Rückschluss auf das Betriebssystem geben[12]. Installierte Schriftarten und Plugins werden bislang überwiegend durch JavaScript erfasst. Anfang 2018 wurde allerdings demonstriert[13], wie installierte Schriftarten auch ohne Einsatz von JavaScript erfasst werden können.

Spezielle Browserfunktionen

Olejnik et al. [138] zeigen zudem, wie Informationen über den Batteriezustand (bei Mobilgeräten) für Fingerprinting genutzt werden können. Aus diesem Grund ist die Batterie-Status API seit Mozilla Firefox 52 deaktiviert[14].

Cross-Browser Tracking

Boda et al. [24] zeigen, wie ein Tracking über Browsergrenzen hinweg durchgeführt werden kann. Dabei werden zum Erstellen des Fingerabdrucks nur Merkmale verwendet, die sich zwischen verschiedenen Browsern nicht unterscheiden. Während bei der User-Agent Angabe im HTTP-Protokoll eine Veränderung beim Browserwechsel zu erwarten ist, gilt dies nicht für Angaben wie den Standort, die IP-Adresse, das Betriebssystem, die Bildschirmauflösung oder die verfügbaren Schriftarten.

Identifikation

Mit FPDetective entwickelten und veröffentlichten Acar et al. [1] 2013 ein Framework zur Erkennung von Fingerprint-Verfahren. Die Autoren sehen Fingerprinting als ein schwerwiegendes Problem an, da sich die Detektion als schwierig erweist und es bisher nur wenige Schutzmaßnahmen gibt. Mit einer Analyse von 1 000 000 Webseiten wurde gezeigt, dass der Einsatz von Fingerprint-Verfahren verbreiteter ist, als in vorherigen Studien angenom-

12 http://www.whatsmyua.info/, abgerufen am 13.03.2018
13 https://github.com/jbtronics/CrookedStyleSheets, abgerufen am 13.03.2018
14 https://www.heise.de/newsticker/meldung/Datenschutzbedenken-Mozilla-entfernt-Akku-Fingerprinting-aus-Firefox-3405099.html, abgerufen am 18.01.2018.

men wurde. Die Autoren unterscheiden verschiedene Techniken, die beim Fingerprinting Verwendung finden und die Erkennung verbessern können:

- Einsatz von JavaScript,
- Nutzung von Plugins und
- Browsererweiterungen.

Einsatz von JavaScript

Die Vorteile der Nutzung von JavaScript für aktives Fingerprinting sind offensichtlich. Auch ist leicht einzusehen, dass Existenz und Version zusätzlich installierter Plugins (wie z. B. Java[15]) eindeutige Merkmale liefern können. Mowery et al. [125] zeigen, wie durch einen Seitenkanal-Angriff Informationen der genutzten Browsererweiterung NoScript erhoben werden und so das Ergebnis des Finterprintings verbessert werden kann. Mowery und Shacham [124] zeigen außerdem, wie mit HTML5 ein Fingerprint der Hardware erzeugt werden kann, die für die grafische Darstellung (Rendering) zuständig ist. Auf diese Weise ist eine Erkennung der Grafikkarte möglich. Mulazzani et al. [126] verwenden JavaScript für eine schnelle und eindeutige Identifizierung von über 150 Browsern, wodurch eigene Modifikationen des User-Agent im HTTP-Protokoll wirkungslos werden.

FP-Toolkits

Nikiforakis et al. [133] analysieren in einer Veröffentlichung von 2013 vier verschiedene Fingerprint-Frameworks: Panopticlick, BlueCava, Iovation ReputationManager und ThreatMetrix. Panchenko et al. [140] zeigen, wie mit Fingerprint-Verfahren Nutzer in anonymisierten Netzwerken (hier: Tor) identifiziert werden können und die Erkennungsrate von 3 % auf 55 % zu steigern.

7.4.2 *Passive FP-Verfahren*

Passive Erkennung

Anders als bei aktivem Fingerprinting findet in der passiven Variante keine Provokation des Browsers zur Preisgabe der Informationen statt. Aus diesem Grund kann die passive Variante auch nicht detektiert werden. Es zeigt sich, dass auch mit wenigen Browserinformationen bereits ein Fingerprinting durchgeführt werden kann. Hierzu führten Yen et al. [205] 2012 eine Studie in Zusammenarbeit mit Hotmail und Bing durch. Die Ergebnisse werden in der Studie B in Abschnitt 8.2.5 näher betrachtet.

Verhaltensanalyse

Bei der Identifizierung über ein Fingerprinting-Verfahren werden nur die Informationen berücksichtigt, die der Browser bei dem Besuch einer Webseite übermittelt. Darüber hinaus kann auch das Verhalten des Nutzers über mehrere Webauftritte hinweg analysiert werden. Olejnik et al. [137] betrachten das Nutzerverhalten von 368 284 Personen und zeigen, wie in 97 % der Fälle die Internetnutzer nach dem Besuch von nur vier Webseiten identifiziert werden können. Die Autoren schlussfolgern, dass die Browsinghistorie wie ein biometrischer Fingerabdruck behandelt werden muss.

Studien

Da passive Verfahren nicht detektieren werden können existieren keine

15 https://docs.oracle.com/javase/6/docs/technotes/guides/jweb/deployment_advice.html, abgerufen am 18.01.2018.

quantitativen Analysen. Ab Abschnitt 8.2 der Studie B wird demonstriert, wie dessen Einsatz bei funktionalen Angeboten (Videos, Schriftarten, etc.) zu einem zusätzlichen Tracking führen kann.

7.5 Übersicht zu Schutzmassnahmen

Überblick

Bisher wurden verschiedene Verfahren zur Umsetzung von Web-Tracking betrachtet. Es stellt sich die Frage, auf welche Weise sich Nutzer vor Web-Tracking schützen können. Die Sichtung der verwandten Arbeiten zeigt, dass Schutzmechanismen in vier Kategorien eingeteilt werden können:

- Schutz durch Blockierung,
- Schutz durch Kontextmanagement,
- Schutz durch Anbietermitwirkung und
- Schutz durch Werkzeuge.

Diese werden in den folgenden Abschnitten 7.6 bis 7.9 genauer beschrieben.

7.6 Schutz durch Blockierung

7.6.1 *Unterdrückung von Werbung*

Werbung vs. Tracking

Grundsätzlich gilt, dass die Befreiung einer Webseite von unerwünschter Werbung nicht mit einem Schutz vor Tracking einhergeht. Ein Werkzeug, welches für ein Ausblenden der Werbung sorgt, muss nicht zwangsweise die darunterliegende Datenübertragung blockieren. Vielmehr können unerwünschte Nebeneffekte der Blockierung vermieden werden, indem die Datenübermittlung gestattet wird und lediglich das Ergebnis, also das werbetragende Element der Webseite, nicht angezeigt wird. Diese Nebeneffekte treten genau dann ein, wenn der Webseitenbetreiber Prüfmechanismen einsetzt, welche Nutzer mit Werbeblockern explizit von den Angeboten der Webseite ausschließen.

Personalisierte Werbung

Obwohl nicht jede Form der Werbung auf vorheriger Profilbildung basieren muss, ist deren Einsatz immer häufiger zu beobachten. Der erwartete Mehrwert ist höher und der Einsatz deshalb wahrscheinlicher. Mayer und Mitchell [115] präsentieren drei verschiedene Formen des Handels mit Werbung (Direct Buy, Advertising Networks, Advertising Exchanges). Agarwal et al. [3] unterscheidet zwischen „Online Behavorial Advertising“ (OBA) und Third-Party-Tracking (TPT). Bei OBA handelt es sich um Dienste, die Daten zur Profilbildung sammeln und diese zum Einblenden von Werbung nutzen. Bei TPT handelt es sich um Dienste, welche eine Datensammlung vornehmen, darüber hinaus jedoch nicht in Erscheinung treten. In der Studie wird durch eine Befragung von 53 Nutzern festgestellt, dass OBA als ein deutlich größeres Problem angesehen wird als TBT:

> „A large number of users in our study reported being more concerned about seeing embarrassing advertisements online

than about their browsing history being tracked by third parties [...] “

Quelle: [3, S. 10].

Sichtbarkeit

Libert [107] berichtet, dass nur 2 % der Tracker im Web über Werbung sichtbar werden. Die verbleibenden 98 % arbeiten im Hintergrund und sind nicht für den Nutzer sichtbar. Es ist zu vermuten, dass die gewonnenen Informationen überwiegend für Werbezwecke verwendet werden. Es verdeutlicht, dass die reine Unterdrückung von Werbung keinen hinreichenden Schutz vor Tracking erbringt.

7.6.2 *Deaktivierung von Third-Party-Cookies*

TP-Cookies

In Abschnitt 7.3 wurde bereits gezeigt, dass Speicherungen zur erneuten Identifikation des Webseitenbesuchers vielfältig sind. Aus diesem Grund ist die alleinige Betrachtungsweise von Cookies (im Sinne von HTTP-Cookies) nicht zielführend. Gemeint ist also an dieser Stelle eine Markierung des Besuchers in jedem technisch denkbaren Sinne. Um den Besucher vor Cookies von Drittparteien zu schützen, müssen alle Speicherformen berücksichtigt werden.

Deaktivierung von TP-Cookies

Der Frage nach der Schutzwirkung durch eine strikte Policy gegenüber Cookies von Drittparteien sind Roesner et al. [153] nachgegangen. Laut den Autoren ist eine vollständige Blockierung dieser Drittpartei-Cookies aus mehreren Gründen nicht zielführend. Zum einen wird dargelegt, wie unterschiedlich diese Striktheit durch die Browserentwickler umgesetzt wird. Die unterschiedliche Umsetzung führt zu einer unterschiedlichen Schutzwirkung. Die Autoren verdeutlichen zudem, dass eine striktere Policy zu einem Verlust der Funktionalität führen kann: Ein Beispiel sind erwünschte Social Widgets oder Single-Sign-On (SSO) Verfahren.

Klassifikation

Unter Berücksichtigung der in Abschnitt 7.1.1 vorgestellten Kategorisierung werden Third-Party-Cookies der Kategorie Vanilla (B) zugesprochen. Im Fall von Google-Analytics, welches in die Kategorie Analytics (A) fällt, findet ebenfalls eine Verfolgung durch eine Drittpartei (Google bzw. Alphabet) statt. Dabei wird auf Third-Party-Cookies verzichtet. Google-Analytics setzt ein HTTP-Cookie mithilfe einer Skriptausführung als First-Party-Cookie einer besuchten Webseite. Auf diesem Weg findet eine eindeutige Identifizierung durch Google statt ohne auf Third-Party-Cookies zurückgreifen zu müssen.

DoubleKeyed-Cookies

Als eine Lösung zwischen Akzeptanz und Blockierung können DoubleKeyed-Cookies angesehen werden, wie sie insbesondere im TOR-Browser Einsatz finden (vgl. Stopczynski et al. [176]). Dabei wird ein Cookie nicht nur für eine Domain gesetzt und stets übermittelt, sondern die aufgerufene Webseite wird ebenfalls als Schlüsselwert gespeichert. Infolgedessen ist das Cookie nur für eine Drittpartei in Verbindung mit genau einer Webseite gültig. Es wird nicht an die Drittpartei übermittelt, wenn diese auf anderen

Webseiten eingebunden ist. Diese Strategie lässt sich ebenfalls auf weitere gespeicherte Daten (z.B. den Cache) übertragen.

7.6.3 *Blockierung von Popups*

Popups

Ein Popup, also ein Seitenaufruf in einem neuen Browserfenster oder -tab, gilt dabei als Aufruf einer Erstpartei. Die Verwendung von Popups kann dazu dienen, die Einschränkungen von Drittpartei-Cookies zu umgehen, wie sie im vorherigen Abschnitt 7.6.2 beschrieben wurden. Über eine Anpassung der GET-Parameter des aufgerufenen Popups können Informationen vom Embedder (Webseite, die das Popup auslöst) an den Tracker (Verantwortlicher des Popups-Inhaltes) weitergegeben werden.

Popupblocker

In modernen Browsern ist bereits ein Schutz vor Popups vorgesehen[16], wie Abbildung 7.3 am Beispiel Mozilla Firefox 54.0 zeigt. Dieser besteht darin, das automatische Öffnen von neuen Browserfenstern und -tabs nur dann zuzulassen, wenn dies durch ein Mausklick initiiert wurde, sofern nicht eine Ausnahme für die jeweilige Webseite vorgesehen wurde. Dies stellt allerdings nur einen schwachen Schutz dar, weil eine Interaktion mit einer Webseite häufig mit Mausklicks, z.B. um einen Link zu folgen, verbunden ist. So wird der Besucher nur beim ersten Öffnen der Webseite vor Popups geschützt.

7.6.4 *Verhinderung von Skriptausführung*

Entwicklung

Nikiforakis et al. [132] zeigen über eine Analyse von 10 000 Webseiten den Anstieg von JavaScript-Einbettungen zwischen den Jahren 2000 bis 2010. Dabei werden die extensive Nutzung von extern bereitgestelltem JavaScript-Code und der stetige Zuwachs neuer Skripte deutlich. Auch Trackingmethoden setzen häufig eine Ausführung von JavaScript auf dem zu trackenden Rechner voraus: Bujlow et al. [28] listet 30 Trackingarten auf, wobei 13 davon auf JavaScript basieren.

7.7 SCHUTZ DURCH KONTEXTMANAGEMENT

Kontextbildung

HTTP-Anfragen im Web stehen immer in einem Kontext. Dieser kann prinzipiell durch stets verfügbare technische Attribute wie z.B. die Absenderadresse (IP) gebildet sein. Auch der Zeitpunkt der Anfrage kann entscheidend sein. Übermittlung weiterer Attribute im Zuge der Client/Server-Konversation schaffen einen noch eindeutigeren Kontext. Beispiele hierfür sind die Übermittlung eines Cookies oder die Nutzung von temporär gespeicherten Inhaltsdaten (Cache).

Schutz des client-state

Trackingverfahren zielen im Allgemeinen darauf, einen solchen Kontext,

16 https://support.mozilla.org/en-US/kb/pop-blocker-settings-exceptions-troubleshooting, abgerufen am 03.01.2017.

als client-state bezeichnet [153], möglichst lange aufrecht zu erhalten[17]. Speicherbasierte Trackingverfahren, beschrieben in Abschnitt 7.3, unterstützen den Tracker bei diesem Ziel. Nur wenn alle beeinflussenden Faktoren bekannt sind, kann durch gezielte Manipulation diese Kontextbildung unterbunden werden.

Faktoren

Dabei muss eine unüberschaubare Vielfalt an kontextbildenden Faktoren berücksichtigt werden, die durch ständige Neu- und Weiterentwicklungen im Browserbereich vergrößert wird.

7.7.1 *Löschen von Cookies, Cache und Browserverlauf*

Cookielöschung

Die Löschung bestehender Cookies, des Caches und der Historie zielt insbesondere auf eine Einschränkung des Trackings auf Basis von speicherbasierten Verfahren ab, wie sie in Abschnitt 7.3 beschrieben wurden. Die Wirksamkeit der Löschung ist in Hinblick auf Cookie-Respawning 7.3.4 kritisch einzuschätzen, sofern nicht alle Speicherformen berücksichtigt werden.

Schutzwirkung

Die Frage nach der Wirksamkeit lässt sich nur schwer beantworten, da diese von der Implementierung des Browsers abhängig ist. Bei dem Operational Cache (Abschnitt 7.3.3) handelt es sich um eine Speicherung, die üblicherweise der Konfiguration des Browsers zugeschrieben wird und keinen direkten Bezug zu Nutzeraktivitäten hat. Infolgedessen muss für jeden Browser und mit jeder neuen Browserversion geprüft werden, ob tatsächlich eine vollständige Löschung vorgenommen wurde. Einen Schutz vor der Verfolgung durch Fingerprint-Verfahren (Abschnitt 7.4) wird ein solcher Bereinigungsvorgang voraussichtlich nicht bewirken.

7.7.2 *Private Browsing*

Nutzen

Der private Modus wurde entwickelt, um Speicherungen während der Nutzung zu unterbinden[18]. Auf diese Weise soll das Anlegen einer Historie verhindert werden, die von anderen Systemnutzern betrachtet werden könnte.

Funktion

Neben der Verhinderung von Seiteneffekten auf dem System wird nach Aktivierung des privaten Modus auch der Zugriff auf den üblichen client-state unterbunden. Da eine Wiedererkennung des Nutzers auf Webseiten auf dem üblichem Weg (überwiegend HTTP-Cookies) somit ausgeschlossen ist, wird der Anschein der Anonymität erweckt.

Schutz

Nutzer könnten missverständlich annehmen, dass dieser Modus einen Schutz durch Beobachtung von außen (Tracker) bewirkt. Roesner et al. [153] stellen klar, dass der Fokus dieser Funktion in dem Verhindern langfristiger Speicherungen auf dem System zu verhindern liegt. Sie ist jedoch kein Trackingschutz: Eine Vermutung, die dahingehend gestützt wird, dass der private Modus anfänglich für cachebasierte Trackingverfahren anfällig war,

17 Stabilitätsanforderung nach Abschnitt 7.1.3.

18 https://support.mozilla.org/de/kb/privater-modus, abgerufen am 28.01.2018.

wie von Ayenson et al. [9] gezeigt wird. Sowohl Bujlow et al. [28, S. 10] als auch Boda et al. [24, S. 14] zeigen, dass der private Modus keinen Schutz vor Tracking durch Fingerprint-Verfahren bietet:

> „Therefore, unless the browser reports a unified and uncommunicative attributes, fingerprinting will work in private browsing mode, at least to some extent. “
> Quelle: [24, S. 14].

7.7.3 *Anonymisierung der Netzwerkschicht*

IP-Adresse

In Abschnitt 7.4.2 wurde gezeigt, dass durch passive Fingerprint-Verfahren die Identifizierung von Nutzern möglich ist. Ein wesentlicher Faktor zur Wiedererkennung ist die IP-Adresse [85]. Neben dem Schutz vor Web-Tracking gibt es weitere mögliche Gründe[19] die eigene IP-Adresse zu verbergen bzw. zu wechseln.

Anonymisierung der IP-Adresse

Das Tor-Netzwerk [116, 91] bietet eine solche Möglichkeit und geht auf eine Technik von David L. Chaum [31] aus dem Jahre 1981 zurück. Auf Basis schalenweise eingesetzter Verschlüsselungstechniken wird dabei der wahre Ursprung einer Web-Anfrage verschleiert. Nur wenn Eintritt- und Ausgangsknoten im Netzwerk unter gleicher Kontrolle stehen, ist eine Identifizierung des Nutzers möglich. Um dies auszuschließen, wird ein ständiger Wechsel dieser Knoten durch den Client vorgenommen.

Schutzwirkung

Dabei muss berücksichtigt werden, dass über die IP-Adresse hinaus keine weiteren Identifizierungsmerkmale übermittelt werden. So ist die Nutzung von Proxys oder eines VPN-Tunnels wirkungslos, wie von Boda et al. [24] gezeigt wird, da der Nutzer auf andere Weise (unbeachtet der IP-Adresse) wiedererkannt werden kann (hier: Fingerprinting). Aus diesem Grund sollte bei einer solchen Anonymisierung auf Netzwerkebene auf die Nutzung des üblichen Browsers verzichtet und eine speziell „gehärtete“ Browserversion eingesetzt werden.

7.8 Schutz durch Anbietermitwirkung

Anbietermitwirkung

Bisher wurden ausschließlich Schutzmöglichkeiten betrachtet, die der Nutzer selbst anwenden muss. Dabei wurde den Embeddern und Trackern die Absicht unterstellt, den Nutzer „um jeden Preis“ verfolgen zu wollen. In der Realität zeigt sich jedoch nicht gänzlich ein solch negatives Bild: So kann auch der Webanbieter ein Interesse daran haben, den Webseitennutzer zu informieren und eine Wahlmöglichkeit zu geben.

7.8.1 *P3P*

Beschreibung

Bei P3P handelt es sich um eine W3C Empfehlung[20], die im Jahr 2002 ver-

19 https://www.torproject.org/about/torusers.html.en, abgerufen am 29.01.2018.

öffentlicht wurde, um Angaben zum Datenschutz[21] als maschinenlesbares Format den Besuchern von Webseiten bereitzustellen. Auf diese Weise soll die Basis für eine Selbstregulierung geschaffen werden. Bereitgestellt werden diese in einer Langfassung (im XML Format) oder in einer verkürzten Form als Zeichenkette.

Entwicklung

Leon et al. [102] zeigen 2010 in einer Untersuchung von 33 139 Webseiten, dass bei 11 176 den Besuchern fehlerhafte Angaben übermittelt werden. Aufgrund dieser falschen Informationen bleiben bei 98 % der Webseiten Cookies unblockiert, die bei korrekter Nutzung von P3P als unerwünscht eingestuft worden wären. Die Autoren merken an, dass P3P keine Verlässlichkeit bietet, sofern Webseiten nicht von unabhängigen Stellen überprüft werden. Untersuchungen von Lämmel und Pek [100] zeigen zusätzliche Schwächen bei der Validierung der Sprache.

Ende von P3P

P3P gilt als offiziell eingestellt[22] und wird schrittweise von den Browserherstellern entfernt[23].

7.8.2 *Do Not Track*

Entwicklung

Do Not Track Flag (DNT) ist ein Attribut des HTTP-Headers. Dieser wird bei jeder Anfrage übermittelt und informiert den Webserver (bzw. Anbieter) über den Wunsch des Nutzers, kein Tracking durchzuführen. Diese Erweiterung des HTTP wurde vom W3C offiziell standardisiert[24]

Schutzwirkung

Bei einer Messung von Acar et al. [2], in der 3 000 Webseiten unter verschiedenen Bedingungen angefragt wurden, zeigten sich bei der Einbettung von Drittparteien und bei Cookie-Syncing nur geringe Änderungen[25]. Balebalko et al. kommen nach einer Messung im Jahr 2012 zum gleichen Ergebnis:

> „In our case study, Do Not Track headers did not seem to limit behavioral targeting of ads. Unfortunately, there are currently millions of Firefox users with DNT enabled who might expect it to have some impact. “
> Quelle: [11, S. 9].

Krishnamurthy und Wills [97] erkennen in DNT keinen Nutzen, sofern keine gesetzlichen Bestimmungen die Umsetzung verpflichten und eine Überwachungsstruktur etabliert wird.

---

20 https://www.w3.org/P3P/, abgerufen am 18.01.2018.

21 In Form einer Datenschutzerklärung bzw. einer Privacy Policy.

22 „P3P Work suspended.“ Quelle: https://www.w3.org/P3P/, abgerufen am 18.01.2018.

23 „The Platform for Privacy Preferences 1.0 (P3P 1.0) is obsolete in Windows 10 (Microsoft Edge and all modes of Internet Explorer 11 for Windows 10).“, Quelle: https://msdn.microsoft.com/en-us/library/mt146424(v=vs.85).aspx, abgerufen am 18.01.2018

24 https://www.w3.org/TR/tracking-dnt/, abgerufen am 29.01.2018.

25 „enabling Do Not Track only reduced the number of domains involved in synchronization by 2.9% and the number of IDs being synced by 2.6%.“, Quelle: [2, S. 9].

7.8.3 *Opt-Out Cookies*

Widerspruch

Opt-Out bedeutet im Allgemeinen, dass von einer grundsätzlichen Einwilligung ausgegangen wird, sofern dieser nicht ausdrücklich widersprochen wurde. Für einen solchen Widerspruch käme der DNT-Flag zwar in Betracht, findet jedoch in der Praxis keine Berücksichtigung.

Widerspruch durch Cookies

Stattdessen bieten vereinzelt Tracker an, den Widerspruch mithilfe von Cookies zu speichern. 94 unterstützende Unternehmen haben sich zu der Vereinigung NAI[26] (Network Advertising Initiative) zusammengeschlossen und ermöglichen den Nutzern eine zentrale Durchführung des Widerspruchs. Sofern mitgesendet, verpflichtet sich der Tracker, die übermittelten Daten nicht zu verarbeiten. Mayer und Mitchell [115] beschreiben als grundsätzliche Problematik, dass bei Löschung der Cookies (zum Schutz vor Tracking) ebenfalls diese Willenserklärung verloren geht.

Schutzwirkung

Acar et al. [2] messen den Effekt von Opt-Out-Cookies bei Besuch von 100 000 Webseiten und stellen fest, dass dessen Wirksamkeit geringer als durch die Löschung von Third-Party-Cookies (Abschnitt 7.6.2) ist. Ebenfalls wurde keine Veränderung des Trackings durch Fingerprint-Verfahren festgestellt und Cookie-Respawning (Abschnitt 7.3.4) wurde weiterhin unverändert durchgeführt. Balebako et al. [11] stellen zwar keine Veränderung bei der Erhebung von Daten fest, jedoch des Werbeinhalts. Den Autoren zufolge entspricht dies dem Versprechen der Tracker, Daten zwar zu erheben, aber nicht für Werbezwecke zu nutzen. Bilenko et al. [21] sehen allerdings neben der Werbung weitere Nachteile, die allein aus der reinen Datensammlung entstehen können.

Bewertung

Opt-Out-Cookies müssen sowohl in ihrer Nutzbarkeit und in Bezug auf ihre Wirkung kritisch bewertet werden. Bereinigungsmaßnahmen führen auch zur Löschung des Opt-Out-Cookies. Datensammlungen werden nicht reduziert, sondern nur die unmittelbare Nutzung für Werbezwecke verhindert.

7.8.4 *Anonymisierungsoptionen*

aip-Flag

Zusätzliche Anonymisierungsoptionen sollen einen datenschutzkonformen Einsatz ermöglichen. Beispielhaft dafür ist die Nutzung der „aip“-Option[27] bei Google-Analytics oder der erweiterte Datenschutzmodus bei YouTube[28].

Schutzwirkung

Im Vertrag[29] zwischen dem Webseitenbetreiber und Google bzgl. der datenschutzkonformen Nutzung von Google-Analytics (Auftragsverarbeitung/Auftragsdatenverarbeitung) wird diese Funktion wie folgt beschrieben: „Bei aktivierter IP Maskierung wird das letzte Oktett der IP Adressen vor

26 http://optout.networkadvertising.org/, abgerufen am 29.01.2018.

27 https://developers.google.com/analytics/devguides/collection/protocol/v1/parameters, abgerufen am 30.01.2018.

28 https://support.google.com/youtube/answer/171780?hl=de, abgerufen am 30.01.2018

29 https://static.googleusercontent.com/media/www.google.com/de//analytics/terms/de.pdf, abgerufen am 30.01.2018.

Speicherung und etwaigen Backups gelöscht“. Es bleibt offen, in welcher Weise die Information zuvor verarbeitet wurde. Der Einsatz von Anonymisierungsoptionen ist wenig verbreitet: Mayer und Mitchell. [115] stellten bei einer Analyse von 10 000 Webseiten im Jahr 2011 fest, dass nur in 63 von 4 861 (1,3 %) Fällen die Anonymisierungsoption von Google-Analytics gesetzt war.

7.8.5 *Zwei-Klick-Lösungen*

Zwei-Klick-Lösungen

Um eine direkte Übermittlung von Daten durch Einbettungen zu verhindern, werden so genannte Zwei-Klick-Lösungen eingesetzt. Der Abruf von Drittanbietern erfolgt erst dann, wenn der Nutzer dies durch eine Betätigung der Schaltfläche initiiert hat. Allerdings ist die rechtliche Konformität zweifelhaft[30], weil Art und Umfang der übermittelten Daten weiterhin nicht abschätzbar sind. Aus technischer Sicht handelt es sich zwar um einen wirkungsvollen Schutz vor automatischer Weitergabe von Daten, allerdings sind diese Lösungen überwiegend nur für Soziale Netzwerke verfügbar. Wie mit den quantitativen Studien in Kapitel 8 noch gezeigt wird, werden neben diesen Netzwerken noch viele andere Inhalte von Drittanbietern eingebunden, für die Zwei-Klick-Lösungen nicht verfügbar sind.

7.9 SCHUTZWERKZEUGE

7.9.1 *Systemschutz*

Systemschutz

Im Laufe der Zeit wurden auf akademischer Basis Schutzverfahren entwickelt und publiziert. Mayer und Mitchell [115] stellen fest, dass diese Entwicklungen als Insellösungen keine ausreichende Verbreitung erreichen. Vielmehr müssten die Browserhersteller diese Konzepte aufgreifen und implementieren.

PRIVAD. Guha et al. [67] entwickelten mit Privad ein Werkzeug, welches eine Verarbeitungsschicht zwischen dem Browser und dem Werbenetzwerk etabliert. Ziel ist es, eine Anonymisierung des Nutzers zu ermöglichen. Die Wahl der Werbung soll lokal erfolgen, um die Weitergabe von Identifikationsmerkmalen zu verhindern. Ein vergleichbarer Ansatz wurde auch von Reznichenko et al. [152] präsentiert.

ADNOSTIC. Toubiana et al. [183] schlagen durch Adnostic ebenfalls eine lokale Selektion von Werbung vor und beschreiben die Selektionsmethodik. Darüber hinaus werden einige allgemeinere Umstellungen vorgeschlagen, welche die Privatheit verbessern.

REPRIV. Das von Fredrikson und Livshits [59] entwickelte Werkzeug teilt die Interessen des Nutzers mit Drittparteien nur dann, wenn dieser

30 https://www.e-recht24.de/artikel/facebook/10081-urteil-abmahnung-facebook-like-button.html, abgerufen am 30.01.2018.

ausdrücklich zustimmt. Damit wird die Einwilligung vom Benutzer zentralisiert und muss nicht für jede Drittpartei neu erteilt werden.

7.9.2 *Selbstschutz*

Schutztools

Der Mozilla Firefox Browser listet bei der Suche nach dem Stichwort „Tracking“ insgesamt 961 Ergebnisse auf (Stand: Januar 2018). Es ist offensichtlich, dass an dieser Stelle nur eine Auswahl an Tools präsentiert wird und keiner vollständigen Auflistung entspricht. Aufgrund der Erkenntnisse aus Abschnitt 7.6.1 wurden Werbeblocker nicht näher betrachtet.

Selbstschutz

GHOSTERY. Ghostery [62] ist ein Anti-Tracking Werkzeug, welches als Erweiterung in den Browser integriert werden kann. Der Schutz wird über eine interne Datenbank[31] umgesetzt, welche eine Erkennung und eine passende Unterdrückung ermöglicht. Dies erfordert die manuelle Pflege durch den Betreiber. Die Software enthält einen Modus, der erhobene Nutzungsdaten an den Betreiber übermittelt[32].

DISCONNECT. Anders als Ghostery ist Disconnect ein quelloffenes[33] Projekt, welches in der Funktionsweise ähnlich umgesetzt ist. Auch hier die eine manuelle Pflege der hinterlegten Tracker durch den Betreiber notwendig. Nachdem das Projekt anfänglich auf Spenden der Nutzer angewiesen war, ist mittlerweile eine Pro- und eine Premiumversion mit erweiterter Funktionalität erhältlich.

PRIVACY BADGER. Der Privacy Badger[34] wurde von der Electronic Frontier Foundation (eff.org) entwickelt. Anders als bei Ghostery oder Disconnect wird hier eine automatische Analyse der Drittparteien durchgeführt und die erhobenen Informationen zur Generierung von Filterlisten verwendet. Eine manuelle Konfiguration (z. B. Blockierung spezieller Tracker) ist möglich, jedoch nicht zwingend erforderlich.

NOSCRIPT. Die Browsererweiterung NoScript[35] unterbindet Skriptausführungen im Browser (Abschnitt 7.6.4). Auch hier ist ein Black- bzw. Whitelisting möglich, bedarf allerdings einer regelmäßigen Anpassung durch den Benutzer. Da aktive Fingerprint-Verfahren auf die Ausführung aktiver Inhalte angewiesen sind, gilt NoScript als effektiver Schutz vor vielen Arten des Trackings. Es muss berücksichtigt werden, dass der legitime Einsatz von Skripten unterbunden wird. Dieses kann zu Einschränkungen der Funktionalität von Webseiten führen.

31 https://www.ghostery.com/faqs/how-does-ghostery-work/, abgerufen am 30.01.2018.
32 https://www.heise.de/tr/artikel/Die-Geister-die-ich-rief-1890700.html, abgerufen am 30.01.2018.
33 https://github.com/disconnectme/disconnect, abgerufen am 30.01.2018.
34 https://www.eff.org/privacybadger, abgerufen am 30.01.2018.
35 https://noscript.net/, abgerufen am 30.01.2018.

REQUEST POLICY. Bei dieser Erweiterung[36] werden Aufrufe von Drittparteien grundsätzlich unterbunden. Über Black- und Whitelisting können die Inhalte manuell durch den Nutzer selektiert werden. Zwar wird auf diese Weise ein umfassender Schutz durch die Ausspähung von Drittparteien umgesetzt, bedarf jedoch, ähnlich wie NoScript, intensiver Pflege durch den Nutzer.

Evaluierung

Eine Evaluierung von drei Schutzmaßnahmen (Adblock Plus, Ghostery, Privacy Badger) wird von Nithyanand et al. [134] durchgeführt. Auch Englehardt et al. [48] messen die Schutzwirkung durch den Einsatz von Ghostery. Weitere Tools/Werkzeuge finden sich auch bei Bujlow et al. [28, S.19].

36 https://requestpolicycontinued.github.io/, abgerufen am 30.01.2018.

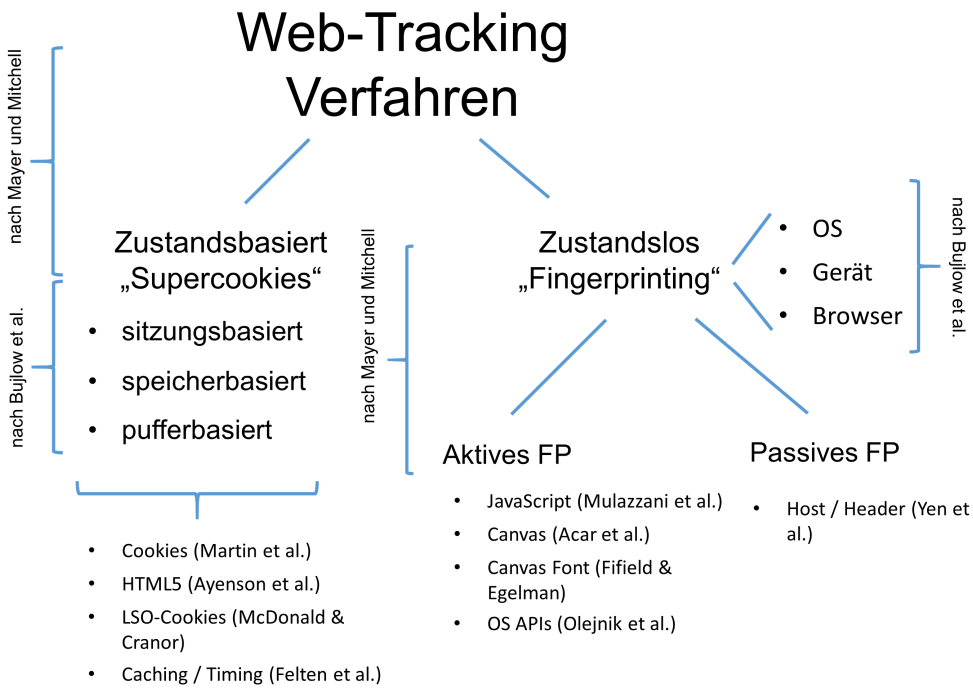

Abbildung 7.1: Übersicht von Trackingarten und -technologien. Verweise: Mayer und Mitchell. [115], Bujlow et al. [28], Mulazzani et al. [126], Acar et al. [2], Fifield und Egelman [58], Olejnik et al. [138], Yen et al. [205], McDonald und Cranor [117], Ayenson et al [9], Martin et al. [114], Felten et al. [53].

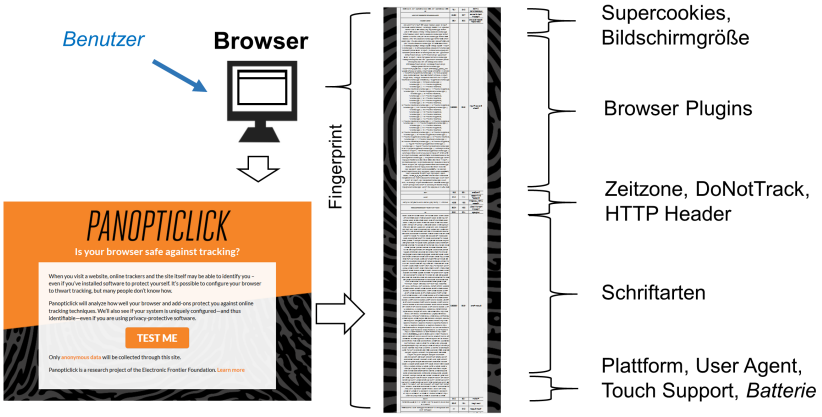

Abbildung 7.2: Übersicht der von Eckersley [44] erhobenen Merkmale zur Generierung eines Fingerabdrucks.

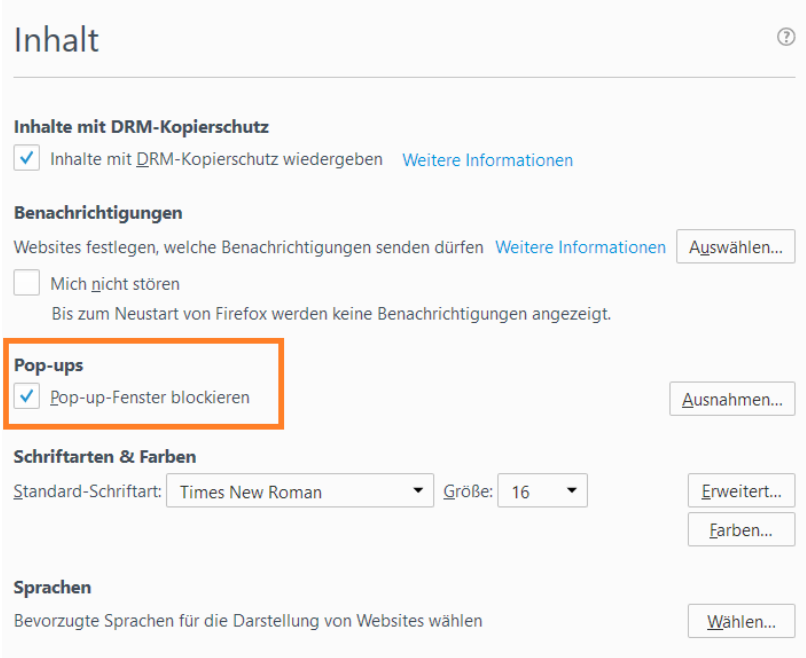

Abbildung 7.3: Popup-Einstellungen im Mozilla Firefox (Version 54.0.)

8

TRACKINGFÄHIGE EINBETTUNGEN AUF DEUTSCHSPRACHIGEN WEBSEITEN

Übersicht

Die in Teil 1 vorgestellte Entwicklung zeigt, dass immer mehr Webseiten Einbettungen vornehmen. Verwandte Arbeiten bestätigen den Eindruck der Zunahme an Drittparteieinbettungen innerhalb der vergangenen Jahre. Im Folgenden werden zwei Studien präsentiert mit dem Ziel

STUDIE A: das „Bewusstsein" der Betreiber mittels Analyse der Datenschutzerklärung zu prüfen und

STUDIE B: technische und rechtliche Implikationen bei Einbettung von Drittinhalten auf Webseiten zu durchleuchten.

Publikationen

Die Studien sind bereits publiziert worden [193, 194, 190]. Die Studie A (Abschnitt 8.1) wurde am 19.07.2015 durchgeführt und Studie B (Abschnitt 8.2) fand am 17.04.2016 statt. Zeitliche Verweise beziehen sich immer auf den Zeitpunkt der jeweiligen Studie.

8.1 STUDIE A: DATENSCHUTZERKLÄRUNGEN VON HOCHSCHULWEBSEITEN

8.1.1 *Kurzübersicht*

Ziele

In dieser Studie wird der Einsatz von Web-Tracking auf Webseiten betrachtet, die üblicherweise nicht auf finanziellen Einnahmen aus dem Webauftritt angewiesen sind. Dabei stand insbesondere die Frage nach dem Bewusstsein von Impressums- und Datenschutzerklärungspflichten bei Betreibern der Hochschulwebseiten im Vordergrund. Über die Handhabung des Datenschutzes, insbesondere beim Einsatz von Tracking, muss der Nutzer gemäß Telemediengesetz (TMG) informiert werden.

Ergebnisse

Eine automatische Prüfung von 207 Hochschulwebseiten, wobei jeweils nur die Hauptseite der Hochschule betrachtet wird, zeigt, dass in 29 Fällen der Dienst Google-Analytics zum Einsatz kommt. Insgesamt fanden sich auf 10 Hochschulseiten Einbettungen von bekannten Trackingdiensten, die in der Datenschutzerklärung der Hochschule nicht berücksichtigt wurden.

8.1.2 *Einleitung*

Situation

Die deutsche Hochschullandschaft umfasst aktuell 426 Hochschulen, die

sich in staatlich, private und konfessionelle Hochschulen eingruppieren lassen. Die Studierendenzahlen schwanken von weniger als 20 bis zu 73 590[1].

Webauftritt

Die Webseite ist das digitale Aushängeschild einer Hochschule, um ihre Studiengänge und Forschungsprojekte zu präsentieren. Im Unterschied zu anderen Webseiten spielen kommerzielle Interessen bei den staatlichen Hochschulen eine untergeordnete Rolle. Es ist anzunehmen, dass die Mitarbeiter sehr frei in der Gestaltung ihrer Webseiten sind und müssen häufig keinen zentralen Vorgaben entsprechen.

Datenschutz

Die Handhabung des Datenschutzes an Hochschulen umfasst viele andere Aspekte (vgl. [201]), die interner Natur sind und weder außen eingesehen noch bewertet werden können. Beispiele sind die Handhabung des Verfahrensverzeichnis oder die Reaktion bei Anfragen zu Datenauskünften.

Ziel

Ziel der Studie ist die Prüfung der Datenschutzerklärungen (kurz: DSE) deutscher Hochschulen. Bei dieser Prüfung soll nachgewiesen werden, dass nicht nur Benutzer bei der Interpretation einer DSE überfordert sind, sondern auch Bildungseinrichtungen nicht in der Lage sind, ihrer rechtlichen Verpflichtung ausreichend nachzukommen.

8.1.3 *Methodik*

Forschungsfrage

Wie in der Einleitung beschrieben ist, wird vermutet, dass Webseiten von Hochschulen nicht hinreichend ihren Pflichten zur Bereitstellung einer vollständigen Datenschutzerklärung nachkommen.

> FORSCHUNGSFRAGE RQ-2: In welchem Umfang findet Web-Tracking auf Webseiten des tertiären Bildungsbereiches statt und wie wird dieses in den Datenschutzerklärungen berücksichtigt?

Vorgehensweise

Zur Beantwortung der Frage muss das Tracking auf den Hochschulwebseiten analysiert und diese IST-Situation mit der SOLL-Situation aus der Datenschutzerklärung abgeglichen werden. Die Studie unterteilt sich in die folgenden Schritte:

- die Betrachtung der verwandten Arbeiten zu dieser Fragestellung,
- eine Entwurfsphase zum Sammeln von rechtlichen Anforderungen an Webseitenbetreiber, die anschließend in technische Anforderungen an die Studie überführt werden,
- die Implementierung des Messwerkzeugs,
- die Wahl einer Testmenge mit anschließender Ausführung der Messung,
- die Präsentation der Ergebnisse,
- die Evaluierung und abschließend
- die Diskussion.

1 Anzahl der Studierenden der FernUniversität in Hagen im Sommersemester 2017 (https://www.fernuni-hagen.de/universitaet/zahlen.shtml, abgerufen am 14.01.2018.)

8.1.4 *Verwandte Arbeiten*

P3P

Leon et al. zeigten 2010 [102] bei einer Analyse von 33 139 Webseiten, dass in 11 176 Fällen fehlerhafte Angaben bei P3P (Platform for Privacy Preferences) vom Webseitenanbieter getätigt wurden. Dies führte dazu, dass die gewünschten Einstellungen des Webseitenbesuchers nicht berücksichtigt werden konnten, bzw. dass bei 98 % der Webseiten mit fehlerhaften Angaben ungewünschte Cookies unblockiert blieben. Balebalko et al. [11] prüften u. a. die Auswirkung des DNT-Flags auf Onlinewerbung. Es zeigte sich, dass dieser nur einen geringen Effekt hat: „DNT seems to have little impact on the number of cookies, which aligns with the industry's low adoption of DNT.", Quelle: [11, S. 8].

Prüfung der DSE

Netzpolitik.org haben bei ihrer Analyse [16] von 35 Webseiten staatlicher Behörden festgestellt, dass ein Viertel dieser Seiten gegen rechtliche Vorgaben verstießen. Bemerkenswert ist, dass auch der Betreiber der Webseite des Bundesverwaltungsgerichts die rechtlichen Anforderungen vernachlässigt haben. Sie bestätigten[2] diesen Fehler und berichtigten ihn.

Weitere Studien

Grundsätzlich sind die quantitativen Analysen zu Web-Tracking aus Abschnitt 3.4 als verwandte Arbeiten anzusehen. Da Hochschulwebseiten im Fokus stehen, ist diese Analyse als Erhebung über „spezielle Gruppe" (gemäß Abschnitt 3.4) einzuordnen.

8.1.5 *Entwurf*

Entwurfsphase

Im ersten Schritt der Entwurfsphase werden die rechtlichen Anforderungen an Hochschulen erfasst. Dabei wird untersucht, was von einer Datenschutzerklärung zu erwarten ist. Auf Basis dieser Erkenntnisse werden im nächsten Schritt die technischen Anforderungen zur Überprüfung abgeleitet.

Rechtliche Anforderungen an den Betreiber

Anwendung des TMG

Die für diese Studie wesentlichen rechtlichen Anforderungen stammen aus dem Telemediengesetz (TMG). Anwendung findet das TMG nur dann, wenn es sich bei einer Hochschule um einen Diensteanbieter im Sinne des § 2 TMG handelt. Dies scheint unstrittig zu sein, wie aus der kommentierten Fassung des TMG hervorgeht:

> „Alleine das nachhaltige Angebot von Telekommunikation mit oder ohne Gewinnerzielungsabsicht genügt. Gemeinnützige Websites ebenso wie Angebote von Bildungseinrichtungen und selbst rein private Homepages sind aufgrund dieser Definition erfasst, da jede auf Dauer angelegte Internetseite das Merkmal der Nachhaltigkeit erfüllt [...] "
> Quelle: [61, Rn. 9-12]

2 https://netzpolitik.org/2015/benutzerverfolgung-durch-staatliche-websites-die-antworten/, abgerufen am 05.01.2018.

Pflichten

Daraus ergibt sich die Verpflichtung, Angaben zum Betreiber und Informationen zur Handhabung des Datenschutzes bereitzustellen. Einschlägig sind hierbei die § 5 TMG „Allgemeine Informationspflichten“ und § 13 „Pflichten des Diensteanbieters“:

Pflichten nach TMG

§ 5 ABS. 1 TMG: „Diensteanbieter haben für geschäftsmäßige, in der Regel gegen Entgelt angebotene Telemedien folgende Informationen leicht erkennbar, unmittelbar erreichbar und ständig verfügbar zu halten:

1. den Namen und die Anschrift, unter der sie niedergelassen sind, bei juristischen Personen zusätzlich die Rechtsform, den Vertretungsberechtigten und, sofern Angaben über das Kapital der Gesellschaft gemacht werden, das Stamm- oder Grundkapital sowie, wenn nicht alle in Geld zu leistenden Einlagen eingezahlt sind, der Gesamtbetrag der ausstehenden Einlagen, [...]“

§ 13 ABS. 1 TMG: „Der Diensteanbieter hat den Nutzer zu Beginn des Nutzungsvorgangs über Art, Umfang und Zwecke der Erhebung und Verwendung personenbezogener Daten sowie über die Verarbeitung seiner Daten in Staaten außerhalb des Anwendungsbereichs der Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr in allgemein verständlicher Form zu unterrichten, sofern eine solche Unterrichtung nicht bereits erfolgt ist. Bei einem automatisierten Verfahren, das eine spätere Identifizierung des Nutzers ermöglicht und eine Erhebung oder Verwendung personenbezogener Daten vorbereitet, ist der Nutzer zu Beginn dieses Verfahrens zu unterrichten. Der Inhalt der Unterrichtung muss für den Nutzer jederzeit abrufbar sein. “

Impressum und DSE

Diese rechtliche Situation verpflichtet den Betreiber sowohl ein Impressum sowie eine Datenschutzerklärung bereitzustellen. Dies insbesondere, wenn die Daten für Werbung, Marktforschung oder zur Erstellung von Nutzerprofilen verwendet werden. Anders als beim Impressum, gibt das TMG nur wenige Vorgaben (§ 13 Abs. 2 TMG) zum Inhalt einer Datenschutzerklärung. Andere Quellen[3] empfehlen den folgenden Aufbau:

- **Zweckbestimmung**: Gründe der Erhebung. Dies impliziert die Nennung von Trackingunternehmen, die zu Analyse- oder Werbezwecken eingebunden werden.
- **Sicherung**: Vorkehrungen zum Schutz der gespeicherten personenbezogenen Daten.
- **Rechte**: Hinweise auf zustehende Rechte z. B. Widerrufs-, Auskunfts- und Löschrecht.

3 OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data

- **Ansprechpartner**: Kontaktinformationen zu einer zuständigen Stelle wie den Datenschutzbeauftragten.

Zusammenfassung

Zusammenfassend kann festgehalten werden, dass auf einer Hochschulwebseite Angaben zum Impressum und zum Umgang mit personenbezogenen Daten in Form einer Datenschutzerklärung erwartet werden können. Sofern Web-Tracking eingesetzt wird, muss auf ein entsprechendes Widerspruchsrecht hingewiesen werden.

Technische Anforderungen der Studie

Nachdem die rechtliche Anforderung formuliert wurde, müssen anschließend die technischen Anforderungen berücksichtigt werden.

Ablauf

Zunächst muss geprüft werden, ob es Angaben zum Impressum und zur Datenschutzerklärung auf der Webseite gibt. Im Anschluss werden alle trackingfähigen Einbettungen durch Mitschnitt der Netzwerkverbindungen erhoben. Bekannte Tracker werden identifiziert und abschließend im Text der Datenschutzerklärung gesucht.

ERKENNUNG VON IMPRESSUMS- UND DATENSCHUTZANGABEN

P3P

Weil Impressums- und Datenschutzangaben keine maschinenlesbaren[4] Bestandteile von Webseiten sind, müssen Informationen dazu in den Inhaltsdaten der Webseite gesucht werden.

Erkennung

Aus der rechtlichen Anforderung diese Informationen leicht erkennbar, unmittelbar erreichbar und ständig verfügbar zu halten, kann von einem Link des Impressums auf der Startseite (Landingpage) ausgegangen werden. In gleicher Weise kann die Webseite auf Links zu „Datenschutz“ oder „Privacy“ analysiert werden.

Suche nach Angaben

Die Ziele dieser Links werden als Angaben zu Impressums- und Datenschutzinformationen angenommen. Auch wenn auf diese Weise noch keine klare Aussage über die Vollständigkeit der Angaben getroffen wird, ist zumindest eine erreichbare Verknüpfung auf der Webseite vorhanden.

Alternative

Falls Impressum oder Datenschutzerklärung über diese Vorgehensweise nicht auffindbar sind, wird dies durch eine Rückmeldung kenntlich gemacht. In diesem Fall können die fehlenden Informationen nachträglich ergänzt werden.

ERKENNUNG VON TRACKING

Übersicht

Zum Zeitpunkt der Analyse stand kein System zur Verfügung, welches eine automatische und zweifelsfreie Erkennung einer Webseitenmenge dieser Größenordnung ermöglichte. Übliche Browsererweiterungen zur Blockierung von Tracking, wie z. B. Ghostery [62], erlauben keine maschinelle Ver-

4 Die Überführung dieser Angaben in eine maschinenlesbare Beschreibungssprache war Ziel des P3P-Projektes, welches jedoch 2007 eingestellt wurde.

Tracker	Domain
Google-Analytics	google-analytics.com
Doubleclick	doubleclick.com
	doubleclick.net
Adition	adition.com
Facebook	facebook.com
	facebook.net
	fbcdn.net

Tabelle 8.1: Zuordnung bekannter Tracker zu deren Domain.

arbeitung der Ergebnisse oder verbieten eine solche Form der Nutzung innerhalb der Nutzungsbedingungen[5].

Automatische Erkennung

Aus diesem Grund wird eine Identifikation bekannter Tracker auf Basis der Netzwerkverbindungen durchgeführt, wie in Teil 1 dieser Arbeit beschrieben wurde. Dabei handelt es sich um trackingfähige Einbettungen, die in Abschnitt 8.2.5 genauer erklärt werden.

Beschränkung

Es muss die rechtliche Unsicherheit berücksichtigt werden, dass nicht jede trackingfähige Einbettung auch in der Datenschutzerklärung genannt werden muss. Als gesichert gilt dies für prominente Trackinganbieter wie Google-Analytics, Doubleclick, Adition und Facebook, in denen eine Profilbildung unbestreitbar ist. Zum Zwecke der Zuordnung findet ein Mapping von Hostnamen auf Trackinganbieter statt, wie in Tabelle 8.1 zu sehen ist.

PRÜFUNG DER DATENSCHUTZERKLÄRUNG

Tracker

Aufgrund der rechtlichen Situation beschränkt sich die Analyse auf vier Trackinganbieter, die nachweislich mit Werbung, Marktforschung oder zur Erstellung von Nutzerprofilen in Verbindung stehen und in der Datenschutzerklärung berücksichtigt werden müssen (§ 15 Abs. 3 TMG).

Automatische Prüfung

Nachdem die Datenschutzerklärung erkannt wurde und trackingfähige Einbettungen den jeweiligen Trackern zugeordnet wurden, wird diese IST-Situation mit der SOLL-Situation der DSE abgeglichen. Dabei werden im Text der DSE nach Hinweisen zu den jeweiligen Trackinganbietern gesucht. Explizit gesucht werden die Stichworte: Google, Alphabet, Doubleclick, Adition, Facebook. Wird eine Verbindung zu Google-Analytics festgestellt, wird die Erwähnung von Google oder Alphabet in der Datenschutzerklärung erwartet.

Manuelle Prüfung

Findet keine Erwähnung statt, wird das Ergebnis manuell geprüft. Auf diese Weise können Falsch-Positive Ergebnisse ausgeschlossen werden.

5 https://www.ghostery.com/about-ghostery/browser-extension-end-user-license-agreement/, abgerufen am 12.01.2018.

8.1.6 *Implementierung*

Erfassung der Drittparteien

Trackererkennung

Als Messwerkzeug zum Erfassen von Einbettungen bekannter Drittparteien diente das zur Analyse der Archivwebseiten eingesetzte Werkzeug (Abschnitt 5.4). Auf die für die Archivwebseiten-Analyse notwendigen Modifikationen wurden verzichtet. Zur Feststellung der Netzwerkverbindungen wurde das PyQt-Framework durch die Einbindung einer eigenen `NetworkAccessManager`-Klasse so modifiziert, dass jede Verbindung durch den Browser protokolliert wird.

Detektion von Drittanbietern

Sobald alle Bestandteile der untersuchten Webseite vollständig geladen sind, ist der Analysevorgang abgeschlossen und eine Auflistung sämtlicher HTTP-Anfragen (`getRequests`) werden ausgegeben. Beispiel: Durch Abruf der Webseite http://www.uni-hamburg.de wurden am Tag der Erhebung neben den Verbindungen zum Webserver der Universität selbst auch Verbindungen zu serveby.flashtalking.com, t4ft.de, google-analytics.com und imagesrv.adition.com registriert.

Weiteres

Verwendet wurde der HTTP User-Agent „User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/534.34 (KHTML, like Gecko) python Safari/534.34“. Dabei handelt es sich um einen typischen Eintrag, der auf aktuellen Windows-Plattformen verwendet wird. Automatische Weiterleitungen werden ausgeführt und berücksichtigt.

Suche und Prüfung von Impressums- und Datenschutzangaben

Parsing

Programmbibliotheken der Programmiersprache Python ermöglichen eine genauere Analyse des Webseiteninhaltes. Nachdem die Hauptseite angefragt wurde, kann mittels BeautifulSoup[6] der Aufbau und dort enthaltene Verweise auf Unterseiten genauer betrachtet werden.

Automatische Suche

Gemäß W3C wird ein Verweis auf eine Unterseite üblicherweise durch ein Anchor Tag beschrieben[7]. Aus diesem Grund werden mit BeautifulSoup diese Verweise aus dem HTML-Dokument extrahiert und Beschreibungstext sowie Zieladresse auf die Stichworte Impressum, Datenschutz und Privacy untersucht. Mehrfachnennungen werden dabei ignoriert. Ergebnis ist eine Liste mit Zieladressen von Impressum und Datenschutzerklärung je Hochschule.

Extraktion

Anschließend werden die Inhaltsdaten extrahiert. Inhaltsdaten sind Informationen, die dem Nutzer beim Abruf der Webseite sichtbar werden und lesbar sind. Enthalten diese, unbeachtet der Groß- und Kleinschreibung, Hinweise auf die vier geprüften Tracker, findet eine solche Erwähnung in der Datenschutzerklärung statt.

6 https://www.crummy.com/software/BeautifulSoup/, abgerufen am 05.01.2018.

7 https://www.w3schools.com/html/html_links.asp, abgerufen am 05.01.2018.

8.1.7 *Ausführung*

Wahl der Testmenge

Gründe

Hochschulen und Universitäten wurden als Ziel der Studie gewählt, weil von diesen ein gewisses Maß an Sorgfalt in Umgang mit dieser Thematik erwartet wird und mit dem Webauftritt keine ökonomischen Ziele verfolgt werden. Im speziellen wurden alle Hochschulen berücksichtigt, die 2.000 oder mehr eingeschriebene Studierende (Stand: Juli 2015) aufweisen. Dies trifft auf 207 von 425 Einrichtungen zu.

Quelle

Weil die Verwaltung von Bildungseinrichtungen den Ländern unterliegt, wurde keine länderübergreifenden Datenquellen gefunden. Infolgedessen wurde die „Liste der Hochschulen in Deutschland“ von wikipedia.org[8] eingesetzt. Eine vollständige Liste befindet sich im Anhang D.2.

Durchführung der Analyse

Ablauf

Bei dieser Analyseform wird nur die Hauptseite betrachtet. Es werden Verbindungen ausgewertet, die durch Aufrufen der Startseite erzeugt werden. Die Unterseiten bleiben von der Analyse des Web-Trackings unberührt. Tag der Erhebung ist der 19.07.2015. Während der Analyse sind keine Fehler aufgetreten.

8.1.8 *Evaluation*

Werkzeugtests

Eine Prüfung des Analysewerkzeugs zur Einbettung von Drittparteien wurde bereits in den Abschnitten 5.4.3 und 6.3.1 durchgeführt. Die Evaluation gilt daher für die hier präsentierte Studie.

Alternativen

Zum aktuellen Zeitpunkt sind keine vergleichbaren Systeme oder Verfahren bekannt, mit denen die hier durchgeführte Automatisierung überprüft werden kann. Die Korrektheit der Suche und Prüfung der Datenschutzerklärung wird durch eine manuelle Überprüfung sichergestellt. Dabei werden zurückgemeldete Negativbeispiele manuell betrachtet und zum Zwecke der Beweissicherung gespeichert.

Manuelle Prüfungen

Falsch-Positive Ergebnisse, dass eine Hochschule den Bestimmungen entspricht, hier jedoch als negativ bewertet wird, können auf diese Weise vollständig ausgeschlossen werden. Falsch-Negative Ergebnisse sind zwar möglich, können jedoch als unkritisch angesehen werden. Während der Durchführung der Evaluation konnten keine fehlerhaften Ergebnisse der automatisierten Suche entdeckt werden.

8 https://de.wikipedia.org/w/index.php?title=Liste_der_Hochschulen_in_Deutschland&oldid=144296605 in der Version vom 23.06.2015, abgerufen am 08.07.2017.

8.1.9 *Ergebnisse*

Existenz von Impressum und Datenschutzerklärung

Impressumssuche

Bis auf eine Ausnahme konnte bei allen untersuchten Webseiten ein Impressum gefunden werden. Bei der Universität der Bundeswehr München[9] konnte das Impressum nicht über die Startseite erreicht werden.

Angaben zur DSE

In 20 der 207 Fälle waren keine expliziten Angaben zum Datenschutz auffindbar: diese Hochschulen sind im Anhang D.2 fett ausgezeichnet. In 137 Fällen wurde die DSE mit dem Impressum zusammengefasst und war unter der gleichen Adresse abrufbar.

Ergebnis

So zeigt sich, dass die Impressumspflicht deutlich intensiver verfolgt bzw. von den Betreibern stärker wahrgenommen wird, als die Verpflichtung zu einer Datenschutzerklärung.

Einbettungen von Drittparteien

Einbettungen

In 105 von 207 Fällen wurden keine externen Verbindungen registriert. In Tabelle 8.2 sind die Serveradressen von Einbettungen zu sehen, die in den verbleibenden 102 Fällen am häufigsten innerhalb der Hochschulwebseiten eingebunden wurden sowie die Anzahl der Hochschulen, die eine solche Einbindung unternommen haben.

Google

In den Analysen zeigte sich, dass 61 der 207 Hochschulwebseiten eine Verbindung zu einem Google-Service bewirkten. Hierbei wurde der Google-Service anhand der Zeichenkette „google“ im Hostnamen bestimmt. Services, die ebenfalls direkt oder indirekt Google angehören, wurden in diesem Fall nicht berücksichtigt: z. B. youtube.com, das 2006 von Google übernommen wurde, sowie doubleclick.com im Jahr 2007. Der Grund hierfür ist, dass nicht in allen Fällen eine vollständige Erfassung geschäftlicher Beziehungen zwischen verschiedenen Trackingunternehmen möglich ist und deshalb zunächst nur die offensichtlichen Verbindungen betrachtet werden können. In Tabelle 8.2 sind die zusätzlichen Google-Services aufgeführt und in Abbildung 8.1 ist der daraus resultierende Graph dargestellt. Dabei stellen die roten Knoten die häufig eingebundenen Drittparteien dar; die restlichen Knoten sind Hochschulen, von denen die ausgehenden Transitionen Einbettungen der jeweiligen Drittpartei darstellen.

Anonymisierung

Tabelle 8.2 zeigt, dass Google-Analytics am häufigsten (29) eingebunden wird. Der Dienst ermöglicht die Nutzung einer Anonymisierungsfunktion, sofern das dafür notwendige Attribut („aip“-Option) vom Webseitenbetreiber gesetzt wurde. Es ist anzumerken, dass die Aktivierung keine Datenübertragung verhindert, sondern lediglich den Wunsch übermittelt, die IP-Adresse nicht vollständig zu berücksichtigen. Dabei entfernt Google-Analytics nach Übertragung der Informationen das letzte Oktett der IP-Adresse des Besuchers (Abschnitt 7.8.4). Aus technischer Sicht ist eine solche nachträgliche Anonymisierung kritisch zu bewerten, da diese weder

9 https://www.unibw.de/, abgerufen am 08.01.2018.

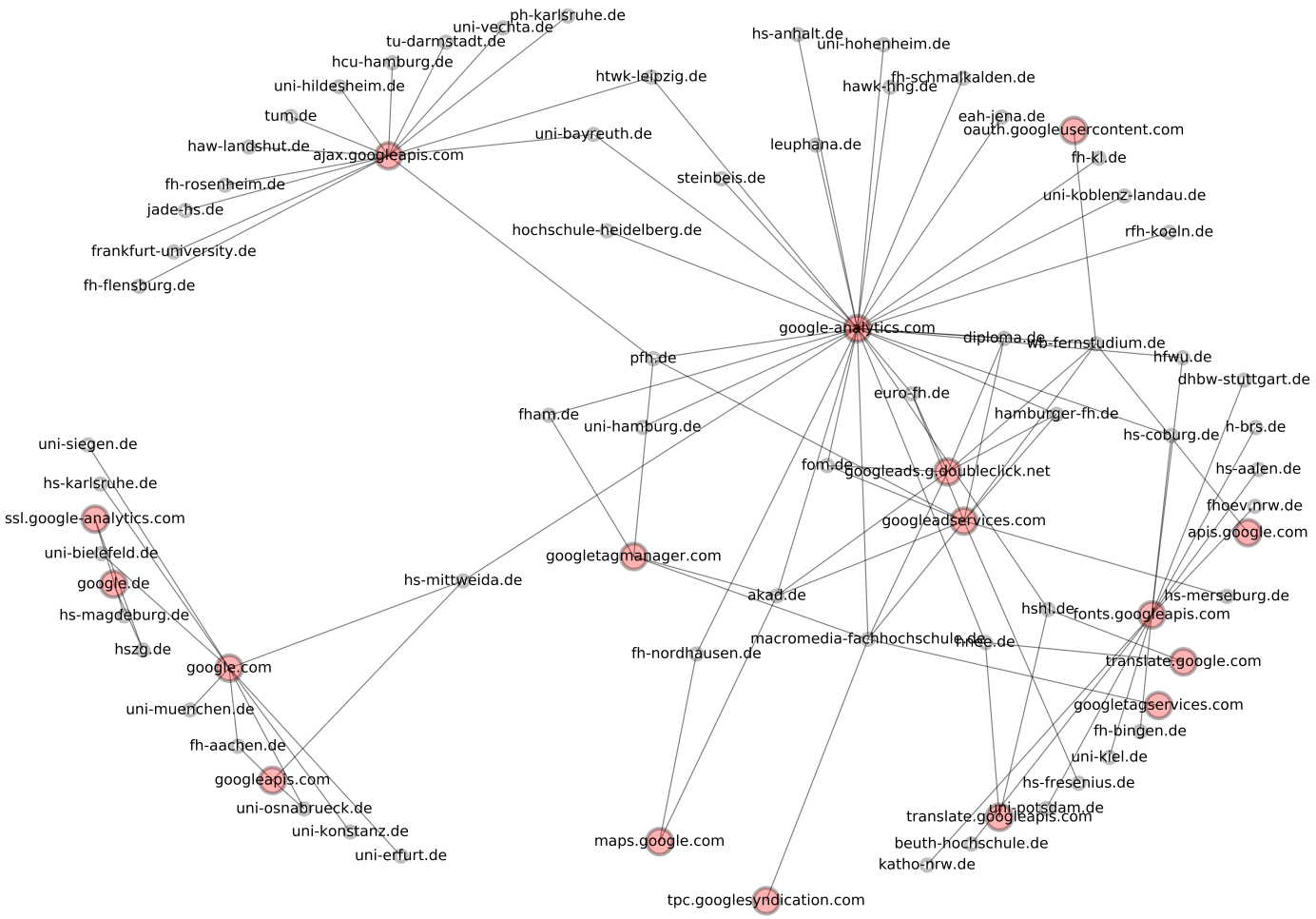

Abbildung 8.1: Eingebundene Google-Dienste (Google im Hostnamen) auf Hochschulwebseiten.

#	Drittpartei	Kategorie	Anzahl
1	google-analytics.com	Web Analytics	29
2	ajax.googleapis.com	Bibliothek	14
3	fonts.googleapis.com	Schriftarten	11
4	www.googleadservices.com	Werbung	10
5	www.google.com	Suchmaschine	9
6	code.jquery.com	Bibliothek	8
7	imagesrv.adition.com	Werbung	8
8	doubleclick.net	Werbung	7
9	s.ytimg.com	Videoportal	6
10	www.youtube.com	Videoportal	6
11	googletagmanager.com	Werbung	4
12	googleapis.com	Bibliothek	3
13	maps.google.com, translate.google.com, ssl.google-analytics.com, translate.google.com, translate.googleapis.com	Verschiedenes	2
14	google.de, googletagservices.com, oauth.googleusercontent.com, tcp.googlesyndication.com		1

Tabelle 8.2: Übersicht der am häufigsten eingebetteten Drittparteien.

nachvollzogen, noch überprüft werden kann. In Tabelle 8.3 ist zu sehen, dass bei 8 von 29 Hochschulen diese Anonymisierungsfunktion nicht genutzt wurde.

Prüfung der Datenschutzangaben

Inhalt der DSE

In 10 Fällen konnte nachgewiesen werden, dass die jeweilige Datenschutzerklärung nicht denen in Abschnitt 8.1.5 genannten Forderungen entspricht und damit als unvollständig gilt. Bei dem Besuch der Webseite wurde mindestens ein Tracker festgestellt, der nicht in der Datenschutzerklärung erwähnt wurde. Die betroffenen Hochschulen sind in Tabelle 8.4 aufgeführt.

Facebook

Obwohl in 83 Datenschutzerklärungen explizit Facebook erwähnt wurde, konnte nur auf zwei Hauptseiten (http://diploma.de, https://www.akad.de) eine Verbindung zu Facebook gefunden werden. Dies ist möglicherweise der fehlenden Analyse der Unterseiten geschuldet.

Zweitprüfung

Die 10 Hochschulen aus Tabelle 8.4 wurden am 08.01.2018 einer erneuten manuellen Prüfung unterzogen. Es zeigte sich eine Verbesserung bei 5 der Hochschulen in dem Sinne, dass entweder die Dienste entfernt (drei Hochschulen[10]) oder vollständig in die Datenschutzerklärung aufgenommen wurden (zwei Hochschulen[11]). In zwei Fällen[12] fand keine Änderung statt.

10 Universität der Künste Berlin, Technische Universität Berlin, Rheinische Friedrich-Wilhelms-Universität Bonn

11 Rheinische Fachhochschule Köln, Universität Koblenz-Landau

12 Hochschule Fresenius (Idstein), Universität Siegen

Hochschule	„aip“-Option
Macromedia Hochschule für Medien und Kommunikation	Ja
Hochschule für nachhaltige Entwicklung Eberswalde	Nein
Steinbeis-Hochschule Berlin	Ja
SRH Hochschule Heidelberg	Nein
Hochschule für Wirtschaft und Umwelt Nürtingen-Geislingen	Nein
AKAD Bildungsgesellschaft (Stuttgart)	Nein
Universität Hohenheim (Stuttgart)	Ja
Hochschule für angewandtes Management (Erding)	Ja
Hochschule für angewandte Wissenschaften Coburg	Ja
Universität Bayreuth	Ja
Diploma Hochschule (Bad Sooden-Allendorf)	Ja
Wilhelm Büchner Hochschule (Pfungstadt)	Ja
Europäische Fernhochschule Hamburg	Ja
HFH Hamburger Fern-Hochschule	Ja
Universität Hamburg	Ja
Private Fachhochschule Göttingen	Ja
HAWK Hochschule Hildesheim/Holzminden/Göttingen	Nein
Leuphana Universität Lüneburg	Ja
Hochschule Hamm-Lippstadt	Nein
Rheinische Fachhochschule Köln	Nein
FOM Hochschule (Essen)	Ja
Fachhochschule Kaiserslautern	Ja
Universität Koblenz-Landau	Ja
Hochschule Mittweida	Ja
Hochschule für Technik, Wirtschaft und Kultur Leipzig	Nein
Hochschule Anhalt (Bernburg, Dessau und Köthen)	Ja
Fachhochschule Nordhausen	Ja
Fachhochschule Schmalkalden	Ja
Ernst-Abbe-Fachhochschule Jena	Ja

Tabelle 8.3: Aktivierung der Anonymisierungsfunktion des Web-Analytic Dienstes Google-Analytics auf Hochschulwebseiten.

Hochschule	Bundesland	Google-Analytics	Doubleclick	Adition	Facebook
FOM Hochschule (Essen)	NRW	J	J		
SRH Hochschule Heidelberg	BW	J			
Hochschule fuer nachhaltige Entwicklung Eberswalde	BB	J			
Steinbeis-Hochschule Berlin	BE	J			
Rheinische Friedrich-Wilhelms-Universitaet Bonn	NRW			N	
Fachhochschule Schmalkalden	TH	J			
Universitaet Koblenz-Landau	RP	J		N	
HAWK Hochschule Hildesheim/Holzminden/Goettingen	NI	J			
Universitaet Bayreuth	BY	J			
Diploma Hochschule (Bad Sooden-Allendorf)	HE	J	J		J
Universitaet der Kuenste Berlin	BE			N	
Hochschule fuer angewandtes Management (Erding)	BY	J	J		
Hochschule Karlsruhe Technik und Wirtschaft	BW			N	
Universitaet Hohenheim (Stuttgart)	BW	J	J		
Wilhelm Buechner Hochschule (Pfungstadt)	HE	J	J		
Hochschule Zittau/Goerlitz	SN	J			
Universitaet Hamburg	HH	J		N	
Hochschule fuer Technik, Wirtschaft und Kultur Leipzig	SN	J			
Hochschule fuer Wirtschaft und Umwelt Nuertingen-Geislingen	BW	J			
Private Fachhochschule Goettingen	NI	J	J		
Philipps-Universitaet Marburg	HE			J	
Duale Hochschule Baden-Wuerttemberg (Stuttgart)	BW				
Macromedia Hochschule fuer Medien und Kommunikation (Muenchen)	BY	J	J		
Fachhochschule Kaiserslautern	RP	J	J		
Hochschule fuer angewandte Wissenschaften Coburg	BY	J			
Europaeische Fernhochschule Hamburg	HH	J	J		
Leuphana Universitaet Lueneburg	NI	J	J		
Ernst-Abbe-Fachhochschule Jena	TH	J			
Hochschule Fresenius (Idstein)	HE		N		
Fachhochschule Nordhausen	TH	J			
Hochschule Anhalt (Bernburg, Dessau und Koethen)	ST	J			
Hochschule Mittweida	SN	J			
Rheinische Fachhochschule Koeln	NRW	N			
HFH Hamburger Fern-Hochschule	HH	J	J		
Universitaet Siegen	NRW			N	
AKAD Bildungsgesellschaft (Stuttgart)	BW	J	J		J
Technische Universitaet Berlin	BE			N	
Hochschule Hamm-Lippstadt	NRW	J			
Hochschule Merseburg	ST		J		
Hochschule Magdeburg-Stendal	ST	N			

Tabelle 8.4: Berücksichtigung von Tracking-Diensten in der Datenschutzerklärung. J: wurde festgestellt und erwähnt / N: festgestellt, nicht erwähnt / sonst: kein Tracking festgestellt.

Auf zwei Hochschulwebseiten[13] zeigte sich hingegen eine Verschlechterung: Weitere Trackingdienste sind hinzugekommen und werden, wie auch die vorherigen, nicht in der DSE erwähnt.

Weiteres

Ein studienübergreifendes Fazit findet in Kapitel 12 statt.

8.1.10 *Diskussion*

Forschungsfrage

Die in Abschnitt 8.1.3 gestellte Forschungsfrage wird im Folgenden beantwortet:

> **In welchem Umfang findet Web-Tracking auf Webseiten des tertiären Bildungsbereiches statt und wie wird dieses in den Datenschutzerklärungen berücksichtigt?**

Situation

In 40 von 207 (19 %) Fällen fand mindestens eine Einbettung von vier prominenten Trackern statt. Bei 61 Hochschulen wurde mindestens eine Verbindung zu einem Google-Dienst registriert (29 %). Bei 20 Hochschulen (9 %) waren keine Angaben zum Datenschutz auffindbar. In 10 Fällen wurde ein Tracking durchgeführt, ohne es in der bestehenden Datenschutzerklärung kenntlich gemacht zu haben. Somit zeigt sich ein nachweisliches Fehlverhalten bei 30 Einrichtungen (14 %).

Werbedienste

Die Verwendung von Werbediensten wie Adition ist auf Webauftritten, insbesondere bei staatlich finanzierten Hochschulen, nur schwierig nachvollziehbar und sollte von den Betreibern überdacht werden. Wie Tabelle 8.4 zeigt, wird der Einsatz von Adition deutlich häufiger verschwiegen als der von Facebook, was u. a. an der stärkeren Sichtbarkeit von Facebook liegen kann.

Facebook

Die Auswertung zeigte darüber hinaus, dass auf keiner Hauptseite eine Einbindung von Facebook vorgenommen wurde, ohne dies innerhalb der DSE zu erwähnen, obwohl in ca. 40 % der Hochschulwebseiten (Impressum, DSE) eine Aussage zu Facebook enthalten ist. Ein möglicher Grund könnte das starke Medieninteresse sein, das bei anderen Trackingdiensten weniger präsent ist. Es ist möglich, dass Facebook vorsichtshalber in die DSE aufgenommen wurde, oder dass die Einbettungen im Laufe der Zeit entfernt wurden.

Ursachen

Die Ursachen der identifizierten Missstände können sein: (1) der sich ständig ändernde rechtliche Rahmen für die DSE, (2) Komplexität und Dynamik von Webauftritten, (3) oft unklare Zuständigkeit bei der Erstellung und Pflege der DSE. Ein weiteres Problem ist der Umfang des Webauftritts einer Hochschule, der häufig mehrere tausend Seiten umfasst. Die DSE muss für alle Seiten gelten, was leicht zu einer Über- oder Unterdeckung führt.

Funktionale Einbettungen

Neben der Einbettung der genannten Trackinganbieter fallen auch Einbettungen zu funktionalen Zwecken auf. Beispiele sind ajax.googleapis.com oder fonts.googleapis.com. Obwohl es sich nicht um klassische Tracker handelt, müssen diese aufgrund der dadurch erzeugten Weitergabe an Daten als

13 Hochschule Magdeburg-Stendal, Hochschule Karlsruhe Technik und Wirtschaft

trackingfähig betrachtet werden. Eine rechtliche und technische Analyse bei Nutzung dieser Drittparteien findet in der Studie B (Abschnitt 8.2) statt.

8.2 Studie B: Drittparteieinbettungen im Gesundheitswesen

8.2.1 *Kurzübersicht*

Motivation

Die Ergebnisse der Studie A (Abschnitt 8.1) zeigen, dass Webseitenbetreiber sich nicht stets über Art und Umfang der eingebetteten externen Komponenten im Klaren sind bzw. die damit einhergehenden rechtlichen Verpflichtungen nicht kennen oder ignorieren. Dabei fiel ein überraschend hoher Umfang an Inklusionen von Drittanbietern auf. Diese können zwar nicht direkt dem Web-Tracking zugesprochen werden, aber sie ermöglichen dieses.

Ziele

In dieser zweiten Studie werden 835 Krankenhaus und Klinikwebseiten untersucht. Zum einen handelt es sich dabei um Webseiten, die von einem breiten Teil der Bevölkerung besucht werden, zum anderen sind Daten zum gesundheitlichen Zustand als besonders sensibel einzustufen und sind deshalb vom Bundesdatenschutzgesetz (§ 3 Abs. 9 BDSG) in besonderer Weise geschützt. Ziel der Studie ist es, das Ausmaß der eingebundenen Drittparteien zu messen und eine rechtliche Bewertung vorzunehmen.

Ergebnisse

Auch hier zeigt sich ähnlich der historischen Entwicklung, dass Google-Analytics dominant ist: 38 % (317) der Webseiten setzen diesen Dienst ein. Sehr deutlich zeigt sich der Einsatz von weiteren Google-Diensten: Schriftarten, Programmbibliotheken, Kartendienste (maps.google.com), statische Inhalte (csi.gstatic.com) und Videos. Während durch Google-Analytics nur ein Drittel der betrachteten Webseiten abgedeckt ist, wird durch die Angebote funktionaler Webseitenbestandteile die Reichweite auf ca. 2/3 (61 %) erhöht. Auch wenn Webseiten solcher Einrichtungen üblicherweise nicht direkt mit Gesundheitsdaten der Besucher in Berührung kommen, ist Google so in der Lage, Besuchsinformationen bei 2 von 3 Krankenhauswebseiten zu erheben.

8.2.2 *Einleitung*

Situation

Die Internetnutzung ist ein wesentliches Mittel der Informationsbeschaffung geworden[14]. Eine vollständige (100 %), mindestens gelegentliche Nutzung der 1 800 Befragten in der Altersgruppe 14 bis 19 Jahre im Jahr 2015. In der Altersgruppe ab 60 Jahre ist dies 2015 bei 50.4 % der Fall. Mit zunehmenden Alter ist die Nutzung des Internets als Informationsquelle für gesundheitliche Themen wahrscheinlich. So z.B. für eine Suche nach Ärzten, Kliniken und Krankenhäusern in der Gegend für sich selbst oder einen nahestehenden Angehörigen. Ebenso auch die Suche nach Behandlungsmöglichkeiten spezieller Erkrankungen, für die es nur wenige Einrichtungen gibt.

Funktionale Bestandteile

Wie bereits in Teil 1 dieser Arbeit gezeigt wurde, sind die Einbettungen von Drittparteien in den vergangenen Jahren angestiegen. Neben den klassischen und bekannten Tracking- und Werbediensten zeigt sich auch eine

14 https://de.statista.com/statistik/daten/studie/36149/umfrage/anteil-der-internetnutzer-in-deutschland-nach-altersgruppen-seit-1997/, abgerufen am 07.07.2017.

Zunahme der Einbettungen, die einen funktionalen Charakter besitzen. Gemeint sind z. B. Anbieter von Schriftarten oder Videos, die mit wenig Aufwand in die Webseite integriert werden können.

Ziele

Solche funktionale Einbettungen finden nur selten Erwähnung in Datenschutzerklärungen, weshalb auf eine Überprüfung verzichtet wird. Es stellt sich jedoch die Frage, ob bei der aktuellen rechtlichen Situation eine Erwähnung zwingend wäre. So soll neben der technischen Betrachtung auch die rechtlichen Aspekte durchleuchtet werden.

Motivation von Einbettungen

Definition

Einbettung ist als ein Sammelbegriff für die Integration von Komponenten in den eigenen Webauftritt zu verstehen. Damit sind Webseitenbestandteile abseits von Texten gemeint: z. B. Grafiken, Videos oder Flash-Applikationen. Im Folgendem wird die Integration von Komponenten betrachtet, die durch einen Dritten zur Verfügung gestellt werden wie z. B. „Like-Button“ von Facebook oder Videos von YouTube.

Einbettungen

Nicht alle Einbettungen dienen dem Web-Tracking. Allerdings sind sie stets mit der Weitergabe der IP-Adresse und weiteren Informationen verbunden und lassen daher eine Nutzung zu diesem Zweck zu. Man könnte annehmen, dass allein die IP-Adresse des Besuchers und die aufgerufene Webseite noch keinen ausreichenden Personenbezug ermöglichen könnten. Diese Annahme erweist sich jedoch dann als falsch, wenn eine „kritische Masse“ an Nutzungsdaten vorliegt: Olejnik et al. [137] beispielsweise zeigen in einem Versuch, dass Benutzer allein an den von ihnen besuchten Webseiten identifiziert werden konnten. Dafür reichte die Kenntnis ihrer IP-Adresse aus, von der aus die verschiedenen Webseiten aufgerufen wurden. Infolgedessen wird davon ausgegangen, dass für die größeren Tracker der Bezug zur Person möglich ist oder zumindest hinreichend bestimmt werden kann.

Gründe für Einbettungen

In vielen Fällen ist eine Unterscheidung, ob es sich um Web-Tracking handelt oder nicht, ein aus technischer Perspektive schwieriges Problem. Auf der einen Seite existieren Einbettungen, welche zur Erfüllung des Zwecks einer Webseite beitragen, z. B. ist die Einbettung eines Produktvideos auf der Webseite eines Warenherstellers. Auf der anderen Seite gibt es Einbettungen, die lediglich aus Geschäfts- bzw. Gewinnerzielungsabsichten getätigt wurden, z. B. durch die Einbettung von Werbung. Zu beachten ist, dass sich die Definition der Funktion nach der Perspektive des Besuchers richtet. Für den Betreiber erfüllt die Werbung eine sehr wichtige Funktion wie z. B. die der Finanzierung. Diese ist, abgesehen von wenigen Ausnahmefällen, üblicherweise nicht der Grund für den Besuch einer Webseite, sondern wird als notwendiges Beiwerk angesehen. Die Unterteilung richtet sich nach der Frage, ob die Komponente untrennbar von der Funktion der Webseite ist, das heißt, ob sie ohne diese ihren Hauptzweck nicht länger erfüllen kann.

Funktionale Bestandteile

Gründe zur Auslagerung von funktionalen Komponenten können sein, dass

(1) dessen externe Lagerung überwiegend Vorteile bringt, oder

(2) aus Bequemlichkeit bzw. unwissentlich eingebunden wurden.

CDN

Zu (1) können u. a. Einbettungen von Videoplattformen (z. B. YouTube) gezählt werden. Die Bereitstellung von Videos an ein größeres Publikum auf dem eigenen Webserver führt zu einem höheren Verbrauch an Datenvolumen und beeinträchtigt damit u. U. die Verfügbarkeit des eigenen Webauftritts oder führt zu zusätzlichen Kosten. Aus diesem Grund bleiben die Daten bei einem externen Anbieter und werden nur in der Form eingebunden, dass für den Betreiber der Webpräsenz keine Last entsteht. Dieser Verbund von Dienstleistern wird auch Content Delivery Network oder Content Distribution Network (CDN) genannt.

CAPTCHAs

Ein anderes Beispiel für (1) sind so genannte CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart), mit der ein Anbieter überprüft, ob der Besucher eine echte Person ist oder nicht. Dies ist insbesondere bei Anmelde-Formularen zu finden. Diese Funktion kann zu Drittanbieter ausgelagert werden, die anschließend eine Rückmeldung über das Ergebnis geben.

Plugins und Themes

Die zweite Gruppe beruht auf dem Unwissen des Betreibers, wie Inhalte des eigenen Webauftritts entstehen. Dieses kann leicht am Beispiel von fertigen Content–Management-Systemen (CMS) erklärt werden. Wird bspw. eine Erweiterung (Plugin) in das CMS eines Webauftritts integriert, sind möglicherweise spezielle Schriftarten enthalten, die prinzipiell vom Betreiber zur Verfügung gestellt werden müssen, damit der Besucher die Webseite in der gewünschten Weise wahrnehmen kann. Aus Gründen der Einfachheit kann allerdings auf bekannte Schriftartendienstleister im Netz verwiesen werden (z.B. fonts.googleapis.com, fonts.net, etc.). Damit bewirkt der Betreiber, dass der Besucher diese Inhalte automatisch nachlädt, ohne sie selbst eingestellt zu haben.

Nichtfunktionale Bestandteile

Nach dem Betrachten funktionaler Komponenten, können nun die nichtfunktionalen Einbettungen näher beleuchtet werden. Beispiele dafür sind:

(3) Social-Media Plugins,

(4) Web-Analytics Dienste und

(5) Werbeanbieter.

Soziale Netzwerke

Unter (3) können bspw. Einbettungen zu sozialen Netzwerken gezählt werden, die in Bildform (z.B. der Facebook „Like-Button“) auf der Seite platziert werden. Sie werden eingesetzt, um die Bekanntheit und Verbreitung der Webseite zu erhöhen.

Analytics

Ein Beispiel für (4) ist der Einsatz von Google-Analytics. Zum Auswerten des Besucheraufkommens eines Webauftritts durch Google-Analytics ist die Einbindung eines Skripts notwendig, welches sich auf einem Google Webserver befindet. Lösungen für den eigenen Betrieb wie bspw. PIWIK sind vergleichbar, jedoch nicht identisch.

Werbung

In den frühen Zeiten des Internets waren durchaus statische Werbeeinblendungen üblich, die der Betreiber im Webauftritt platziert hatte. Heute

findet eine Einbettung von einem Werbeunternehmen (5) statt, das den angezeigten Inhalt selbst bestimmt und so vom Profil des Benutzers abhängig machen kann.[15]

Szenario

Hintergrund

In dem für die Studie erdachten Szenario nutzt eine Person das Internet, um eine für sein bereits diagnostiziertes Leiden ein passendes Klinikum zu finden. Unter der Prämisse, dass keine Suchmaschine verwendet wurde, sondern die Webseite bekannt und direkt in der Adressleiste des Browsers eingegeben wurde, ruft der Besucher die Hauptseite (Landingpage) der Klinik ab. Im Anschluss folgt eine Durchsicht der einzelnen Fachabteilungen. Genauer betrachtet werden die Leistungen zur Lungenkrebsbehandlung, die als dritthäufigste Krebsursache gilt. Einen ersten Eindruck der dort behandelnden Ärzte und zu deren Reputation soll auf diesem Weg ebenfalls gewonnen werden. Darüber hinaus erfolgt eine Durchsicht der laufenden klinischen Studien, falls solche Informationen bereitgestellt werden.

Besonderheiten

Bei einem solchen Szenario ist grundsätzlich anzunehmen, dass die Person bei der Recherche auf einer Klinik-Webseite ein gewisses Maß an Anonymität erwartet. Manche Leiden lassen ggf. auch Rückschlüsse auf die Umstände zu, die zur Erkrankung geführt haben (bspw. Geschlechtskrankheiten). Es ist naheliegend, dass diese sensiblen Informationen nicht mit unberechtigten Dritten geteilt werden sollen.

8.2.3 *Methodik*

Ziel

Ziel der Studie ist es, die technischen und rechtlichen Konsequenzen von Drittparteieinbettungen zu betrachten und den „aktuellen Zustand“ zu erfassen. Diese Zustandserfassung erfolgt durch das Betrachten von Webseiten aus dem Gesundheitswesen – also von Krankenhäusern und Kliniken.

FORSCHUNGSFRAGE RQ-3: In welchem Ausmaß finden Einbettungen von Drittinhalten auf Webseiten im Gesundheitswesen statt und was sind die Konsequenzen?

- RQ-3.1 Was sind die technischen und rechtlichen Konsequenzen von Einbettungen im Webauftritt?
- RQ-3.2 Wie ist die aktuelle Situation auf Webseiten von Krankenhäusern und Kliniken?

Vorgehensweise

Zur Beantwortung der Fragen wird genauer analysiert, welche Daten bei Durch- und Ausführung von Einbettungen übermittelt werden. Es wird untersucht, ob die Identifizierung allein auf Basis dieser übermittelten Daten möglich ist. Ist dies der Fall, muss von der Durchführung eines Trackings auf Basis dieser Daten ausgegangen werden. Nach dem Betrachten verwandter

15 Dieser Abschnitt wurde bereits (vom Autor allein) vorab publiziert [190] und an dieser Stelle übernommen.

Arbeiten werden diese Fragen in Abschnitt 8.2.5 näher behandelt. Um das Ausmaß dieser Einbettungen auf eine bestimmte Gruppe von Webseiten zu bestimmen, werden passende Analyse- und Auswertungswerkzeuge entwickelt.

Struktur

Die Studie wird in die folgenden Abschnitte untergliedert:

1. Analyse der verwandten Arbeiten,
2. technische Betrachtung der Datenweitergabe durch Einbettungen,
3. Analyse der rechtlichen Konsequenzen,
4. Implementierung eines Messwerkzeugs,
5. Wahl der Testmenge und Ausführung der Messung,
6. Aufbereitung und Präsentation der Messergebnisse,
7. Evaluation und abschließend
8. Diskussion.

8.2.4 *Verwandte Arbeiten*

Libert

Libert veröffentlichte 2015 eine Studie mit dem Thema „Privacy Implications of Health Information Seeking on the Web“ [106]. In dieser wurden Einbettungen von Drittinhalten auf Webseiten betrachtet. Die Menge der geprüften Webseiten wurde durch eine Internetsuche nach Stichwörter zu Erkrankungen generiert.

Unterschiede zu Libert

Diese Vorgehensweise zur Generierung einer Testmenge ist nur dann sinnvoll, wenn der Schutz der Privatheit im Fokus steht. Es liegt nahe, dass Internetnutzer zunächst eine Suchmaschine verwenden, um nach Symptomen einer Krankheit zu suchen, um anschließend die Ergebnisse zu sichten und gegebenenfalls Webseiten zu besuchen. Während dieses Vorgangs ermöglicht ein Tracking eine Verbindung zwischen der Person und dem gesuchten Krankheitsbild. Es ist anzumerken, dass dabei auch Webseiten abgerufen werden, deren alleiniges Geschäftsmodell auf Internetwerbung basiert. Webseiten, deren Inhalte möglicherweise durch intensive Recherchearbeiten produziert wurden und aus diesem Grund über Werbung finanziert werden müssen. Anders als bei Libert wird in der vorliegenden Studie davon ausgegangen, dass Krankenhäuser und Kliniken nicht allein auf Einnahmen durch Internetwerbung angewiesen sind.

Weitere

Wie in Studie A sind in der vorliegenden Studie quantitativen Analysen zu Web-Tracking, die in Abschnitt 3.4 beschrieben wurden, als verwandte Arbeiten anzusehen. Die Analyse von Krankenhaus- und Klinikwebseiten ist wie bei Hochschulen als „spezielle Gruppe“ (gemäß Abschnitt 3.4) einzuordnen. Besonders tritt dabei die Arbeit von Yen et al. [205] hervor:

> „We further demonstrate the privacy and security implications of host-tracking in two contexts. In the first, we study the causes of cookie churn in web services, and show that many returning users can still be tracked even if they clear cookies or utilize

private browsing. [...] “
Quelle: [205, S. 1].

Die Studie wird im Entwurfsabschnitt genauer betrachtet und eingeordnet.

8.2.5 *Entwurf*

Datenweitergabe durch Einbettungen

Einbettung

Durch eine Referenzierung auf externe Ressourcen wird bei Abruf durch den Browser eine ggf. neue HTTP-Verbindung aufgebaut. Die Analyse von Yen et al. [205] zeigt, dass eine Profilbildung auf Basis der während dieses Vorgangs übermittelten Informationen möglich[16] ist, die ohne weiteres Zutun bei Abruf einer Ressource übertragen wird. Demzufolge kann Benutzerverfolgung allein auf den Daten basieren, die ein fester Bestandteil der Internetkommunikation sind. Die gleichen Daten, die auch als Folge von Einbettungen übermittelt werden.

passives Fingerprinting

Yen et al. berücksichtigen nur die IP-Adresse sowie die Angaben aus dem HTTP-Header. Darüber hinaus kommen weitere Protokolle in Betracht, die eine Verbesserung der Identifikationsrate ermöglichen. So müssen neben diesen IP- und HTTP-Informationen ebenfalls die dazwischenliegenden oder damit verbundenen Protokolle einbezogen werden.

Protokolle

In Abbildung 8.2 sind die wesentlichen beteiligten Protokolle abgebildet: IP, TCP, SSL/TLS, HTTP. Aus Gründen der Übersicht wurde auf eine Abbildung der vorherigen DNS-Abfrage verzichtet, muss jedoch ebenfalls bedacht werden. In diesem Beispiel greift der Browser auf eine Webseite (R1.html) der Domain I zu (vgl. Label L1). In dieser findet eine Einbettung statt und verweist auf das Content Distribution Network (CDN) der Domain II. Der Browser folgt diesem selbstständig und sendet (Label L2), um den HTTP-Request durchzuführen ein IP-Paket, welches neben TCP- und ggf. TLS- einen HTTP-Header enthält. In diesem Header wird als Referrer der Ursprung der Einbettung angegeben (R1.html). Um festzustellen, in welcher Weise eine Informationsweitergabe bewirkt wird, müssen alle Kommunikationsverbindungen betrachtet werden, die durch die Einbettung ausgelöst werden.

IP (V4) In privaten Haushalten werden IP-Adressen (Version 4 [85]) dynamisch vergeben. Dies bedeutet, dass IP-Adressen nur für eine gewisse, je nach Anbieter abweichende, Zeitspanne dem Anschluss zugewiesen werden. Trotz des Anonymisierungsversuches muss die IP-Adresse als personenbezogenes Datum betrachtet werden[17]. Aktuelle

16 „We show that 60%-70% of HTTP user-agent strings can accurately identify hosts in our datasets. When augmented with coarse-grained IP prefix information, the accuracy can be improved to 80%, similar to that obtained with cookies. User-agent strings combinedwith IP addresses have an entropy of 20.29 bits—higher than that of browser plug-ins, screen resolution, timezone, and system fonts combined [20]“, Quelle: [205, S. 2]

17 Der europäische Gerichtshof (EuGH) erklärt, in einem Urteil vom 06. Dezember 2016 (C-582/14), auch dynamische IP-Adressen als personenbezogen – eine Auffassung, die vom BGH in einem Urteil vom 15.05.2017 bestätigt wird (VI ZR 135/13).

Entwicklungen im Telekommunikationsbereich zeigen, dass die mit dem IP-Wechsel verbundene Zwangstrennung ein Hindernis[18] darstellen kann, z. B. im Fall von IP-Telefonie. Als Reaktion findet je nach Anbieter und Anschlusstyp eine Zwangstrennung nur noch alle 180 Tage statt[19].

Die IP-Adresse liefert also über einen gewissen und aktuell steigenden Zeitraum einen verlässlichen Hinweis auf den Anschluss. Abfragen mit nur kurzen zeitlichen Abständen können mit hoher Sicherheit zusammengefasst werden. Wird an einer Stelle eine genauere Identifikation, beispielsweise durch ein HTTP-Cookie ermöglicht, ergibt sich ein Personenbezug für die vorangegangenen Abfragen.

Bei Einsatz von *Network Address Translation* bzw. *Port Adress Translation* (NAT/PAT) [47] verwenden mehrere Geräte die gleiche Absenderadresse, wodurch ein Personenbezug erschwert[20] wird.

IP (V6) Schon vor dem Jahr 2006 wurde erkannt, dass aufgrund der Überverfügbarkeit von IPv6-Adressen [76] auch langfristige Zuordnungen von IP-Adressen zu Geräten möglich sind. Die automatische Generierung einer IPv6-Adresse via SLAAC (Stateless Address Autoconfiguration) [128] führt dazu, dass die MAC-Adresse Bestandteil der IP-Adresse wird. Diese wird üblicherweise vom Hersteller der Netzwerkschnittstelle fest zugeordnet und ist grundsätzlich als ein unveränderliches Merkmal vorgesehen. Den daraus resultierenden Privatheitsproblemen widmet sich die RFC 4941 [127]. Als Lösung wird die Nutzung eines Hash-Verfahrens zur Generierung einer pseudozufälligen (temporären) Adresse vorgeschlagen: „This document proposes the generation of a pseudo-random sequence of interface identifiers via an MD5 hash." Quelle: [127, S. 8].

Es ist zu berücksichtigen, dass eine solche zufällige Generierung nur im Rahmen des zugewiesenen Netzes möglich ist. Insbesondere bei pseudozufälliger Generierung kann anhand weniger Adressen auf den Adressbereich, also das Subnet, geschlossen werden, der dem jeweiligen Anschluss zugewiesen wurde. Wie bei IPv4 muss bei IPv6 in gewissen Zeitintervallen dem Anschluss ein neuer Adressbereich zugewiesen werden.

Die Nutzung von IPv6, auch mit der beschriebenen SLAAC-Erweiterung, verbessert die Lage nicht. Während bei IPv4 mittels NAT alle Geräte in einem lokalen Netzwerk nur mit einer einzigen IP-Adresse von außen sichtbar sind, ist mit IPv6 eine feingranulare Unterscheidung der Anfragesteller möglich.

TCP/UDP Das Transmission Control Protocol [145] (TCP) und das User Datagram Protocol [144] (UDP) sind die am häufigsten genutzten Trans-

18 https://www.heise.de/newsticker/meldung/Unerwuenschte-Trennung-bei-All-IP-Anschluessen-Telekom-spielt-Software-Korrektur-ein-2514444.html, abgerufen am 14.01.2018.

19 https://www.teezeh.de/2014/01/06/telekom-keine-taegliche-zwangstrennung-mehr-bei-dsl/, abgerufen am 14.01.2018.

20 Dazu auch Yen et al. [205, S. 5], Abschnitt 3.3: Impact of Proxys and NATs

portprotokolle im Internet. Sowohl SSL/TLS als auch HTTP benötigen die speziellen Vorteile von TCP. Das DNS-Protokoll hingegen kann auch auf Basis von UDP transportiert werden.

Bei TCP sind die Felder Time to Live, Window Size, Maximum Segment Size, das „Don't fragment"-Flag und die Optionen SACK und Window Scale typischerweise betriebssystemspezifisch. Die Erkennung von Systemen üblicherweise das Betriebssystem anhand TCP/IP Protokollinformationen ist in der Informationssicherheit bereits eine gängige Praxis (vgl. Greenwald und Thomas [65] sowie Medeiros et al. [118]). Derartige Verfahren kommen u. a. in Portscannern zum Einsatz[21].

Die Verwendung dieser Informationen für Web-Tracking wird von Eckersley [44] angedeutet. Nikiforakis et al. stellen die Nutzung von TCP/IP-Parameter zum Erstellen von Fingerabdrücken im Fingerprint-Framework BlueCava fest, führen diese jedoch nicht genauer aus [133]. Es ist anzunehmen, dass durch die Berücksichtigung dieser Informationen die Erkennungsraten von Yen et al. [205] deutlich gesteigert werden können.

DNS Daniel Dent zeigt seit 2015 in einem Demonstrator[22], wie zwischengespeicherte (cached) DNS-Informationen zur Generierung eines browserübergreifenden Cookies verwendet werden können. Dabei werden mit Besuch der Webseite 32 DNS-Abfragen ausgelöst, wobei speziell präparierte DNS-Server zufällige Antworten zu den festgelegten Hosts (00c.dnscookie.com - 31c.dnscookie.com) liefern. Diese zufälligen IP-Auflösungen stellen die Bits des DNS-Cookies dar. Der Client speichert nun die 32 Antworten des DNS-Servers. Betritt der Nutzer erneut die Webseite, ist eine Auflösung der Hosts nicht erforderlich, da diese Informationen noch zwischengespeichert sind. Je nachdem, welche IP-Adresse des jeweiligen Hosts vom Client kontaktiert wird, gibt der Besucher die einzelnen Bits des DNS-Cookies preis.

Diese vergleichsweise umständliche Vorgehensweise ist mittels IPv6 deutlich einfacher umzusetzen. Bei der Abfrage eines Hosts wird eine zufällige IPv6-Adresse zurückgeliefert und vom System zwischengespeichert. Dieser Zwischenspeicher kann, bei einem Wiederbesuch der Webseite zur Identifizierung genutzt werden.

Es ist anzumerken, dass das Tracken durch die Lebensdauer der DNS-Einträge begrenzt ist. Weil eine kurze Zeitspanne zur Wiederherstellung gelöschter Cookies schon ausreichen kann, muss diese Trackingart berücksichtigt werden.

SSL/TLS Das Secure Sockets Layer (SSL) bzw. Transport Layer Security (TLS) ermöglicht den Schutz von Vertraulichkeit, Integrität und Authentizität der übertragenen Daten. Nach dem Verbindungsaufbau der darunterliegenden Transportschicht sendet im ersten Protokollschritt

21 https://nmap.org/book/osdetect-methods.html, abgerufen am 14.01.2018.

22 http://dnscookie.com/, abgerufen am 15.01.2018.

der Client ein `client_hello`, welches die vom Client unterstützten Cipher Suites enthält. Dabei handelt es sich um eine Liste von kryptografischen Verfahren, die vom jeweiligen Gerät, Betriebssystem und Browser unterstützt werden. Diese können je nach Browser oder Gerät abweichen[23], was ein Fingerprinting ermöglicht oder verbessert. Auch eine Session-ID zur Wiederaufnahme einer unterbrochenen SSL/TLS-Sitzung kann zum Web-Tracking eingesetzt werden[24].

HTTP Die vielzähligen Möglichkeiten auf HTTP-Basis ein Tracking durchzuführen, wurde bereits in Kapitel 7 behandelt. Dies umfasst eindeutige Verfahren wie ein HTTP-Cookie, jedoch auch weniger offensichtliche wie die Verwendung von E-Tags nach RFC 2616 [121] (bereits beschrieben in Abschnitt 4.4.3). An dieser Stelle wird der HTTP-Referrer näher betrachtet. Das Feld diente ursprünglich[25] administrativen Zwecken, z. B. zur Suche fehlerhafter Verlinkungen. Faktisch zeigt sich, dass entweder das Referrer- oder das 2011 hinzugekommene Origin-Feld[26] (nach RFC 6454 [13]) eine Informationsweitergabe bezüglich der besuchten Webseite an den Anbieter der eingebetteten Ressource bewirkt. Nach RFC ist der Webserver nicht gezwungen, das Referrer-Feld auszuwerten. Unbeachtet dessen ist die Standardkonfiguration aktueller Browser[27] die Übermittlung dieser Informationen.

Zusammenfassung

Bei einer Einbettung, wie in Abbildung 8.2 skizziert ist, werden quer durch die Schichten Merkmale zur Identifikation und Informationen zum Nutzungsverhaltens weitergegeben. Aus technischer Perspektive kann eine Nutzerverfolgung auch durch eingebettete funktionale Komponenten durchgeführt werden.

Rechtliche Analyse von Einbettungen

Übersicht

Die rechtliche Einschätzung der Einbettung von Drittparteien wurde von Dr. Laura Schulte von der Universität Bielefeld im Rahmen der Veröffentlichung der Ergebnisse [194] durchgeführt und wird an dieser Stelle als Zitat vollständig übernommen. Auf diese Weise wird die Verständlichkeit der Arbeit ohne die Gefahr einer Verfremdung des Inhaltes sichergestellt.

Beginn des Zitats von Dr. Laura Schulte.

> Datenschutzvorgaben, die bei dem Betrieb von Webseiten relevant werden, begründet das Telemediengesetz (TMG). Gem. § 12 Abs. 1 TMG darf ein Diensteanbieter, namentlich der Webseitenbetreiber,

23 https://technet.microsoft.com/en-us/library/dn786419(v=ws.11).aspx, abgerufen am 15.01.2018.

24 https://trac.torproject.org/projects/tor/ticket/4099, abgerufen am 28.01.2018.

25 „The Referer request-header allows a server to generate lists of back-links to resources for interest, logging, optimized caching, etc. It also allows obsolete or mistyped links to be traced for maintenance.“ Quelle: [121, S. 139].

26 Das Origin-Feld enthält die Adresse der besuchten Webseite.

27 Z. B. Mozilla Firefox http://kb.mozillazine.org/Network.http.sendRefererHeader, abgerufen am 15.01.2018

nur dann personenbezogene Daten erheben und verwenden, wenn und soweit entweder ein spezialgesetzlicher Erlaubnistatbestand hierfür besteht oder der Nutzer in die Verarbeitung wirksam eingewilligt hat.
Vorliegend stehen die Verarbeitung von Daten, die der Identifizierung des Nutzers dienen, Angaben über den Beginn und das Ende der jeweiligen Nutzung sowie Angaben über die vom Nutzer in Anspruch genommenen Telemedien in Rede, mithin Nutzungsdaten i.S.v. § 15 Abs. 1 S. 2 Nr. 1-3 TMG. Als Erlaubnistatbestand kommt entsprechend § 15 Abs. 1 S. 1 TMG in Betracht[1]. Hiernach darf der Diensteanbieter die Daten nur erheben und verwenden, um die Inanspruchnahme von Telemedien zu ermöglichen.
Fraglich ist, ob die Einbettung von Drittinhalten überhaupt eine Erhebung personenbezogener Daten durch den Seitenbetreiber darstellt. Erheben ist das Beschaffen von Daten über den Betroffenen, § 3 Abs. 3 BDSG. Hierfür ist es nicht zwingend erforderlich, dass der Seitenbetreiber Besitz an den Daten bzw. die physische Herrschaft über den Verarbeitungsprozess erhält[2]. Vielmehr genügt es bereits, dass er durch die Einbettung fremder Inhalte Dritten die Möglichkeit verschafft, auf personenbezogene Daten zuzugreifen, und so deren weitere Verarbeitung und Nutzung initiiert[3]. Insoweit stellt das Einbetten von Drittinhalten auch das Beschaffen von Daten durch den Seitenbetreiber dar.
Für den Webseitenbetreiber ist die Kenntnis der IP-Adresse erforderlich, um die Inanspruchnahme des Dienstes zu ermöglichen. Die Erhebung von Daten i.d.S., dass diese Dritten zugänglich gemacht werden, ist grundsätzlich nicht für den Betrieb der Seite erforderlich. Drittinhalte, etwa Schriftarten, können vielmehr vom Seitenbetreiber selbst gehostet werden. Die Einwände, die Einbettung von Drittinhalten sei ein integraler Bestandteil der Funktionsfähigkeit des Internet und das Hosting durch den Seitenbetreiber für diesen mit mehr Aufwand verbunden, vermögen nicht zu überzeugen. Kontrafaktizität bzw. ein erhöhter Aufwand entbinden den Seitenbetreiber nicht von bestehenden datenschutzrechtlichen Verpflichtungen.
Selbst wenn die Drittpartei vorgibt, eine Kürzung der IP-Adresse vorzunehmen – wie Google in Bezug auf seinen Dienst Google Analytics (aip Flag[4]) –, hat dies keine Auswirkungen auf den Personenbezug der Daten in der hier relevanten Phase: Eine etwaige Kürzung bzw. Anonymisierung der IP-Adresse erfolgt aus datenschutzrechtlicher Perspektive zu spät, nämlich erst im Arbeitsspeicher der Drittpartei und damit nach der bereits datenschutzrechtlich relevanten Phase der Erhebung.
Für Zwecke der Werbung, Marktforschung oder zur bedarfsgerechten Gestaltung des Telemediums darf der Webseitenbetreiber selbst Nutzungsprofile unter der Verwendung von Pseudonymen

erstellen, solange der Nutzer dem nicht widerspricht, § 15 Abs. 3 S. 1 TMG. Dieser Erlaubnistatbestand gilt aber nur für den Diensteanbieter selbst, die Verarbeitung personenbezogener Daten durch Dritte wird hingegen nicht erlaubt. Das Vorliegen eines Auftragsdatenverarbeitungsverhältnisses zwischen dem Webseitenbetreiber und der Drittpartei, bei dem ersterer gem. § 11 Abs. 1 S. 1 BDSG als ausschließliches Zurechnungssubjekt von datenschutzrechtlichen Pflichten und Rechten gelten würde, ist auszuschließen. Die zugrundeliegende Konstellation entspricht nicht dem gesetzlichen Leitbild der Auftragsdatenverarbeitung; im Zweifel hat der Webseitenbetreiber weder Kenntnis geschweige denn Kontrolle über die Datenverarbeitung der Drittpartei.

Weitere Erlaubnistatbestände des TMG sind nicht einschlägig. Dennoch kann die Einbettung fremder Inhalte datenschutzrechtlich zulässig ausgestaltet werden. Hierzu bedarf es der wirksamen Einwilligung des Nutzers. Diese muss den verschiedenen Anforderungen des § 13 Abs. 1-3 TMG gerecht werden. Insbesondere muss die Einwilligung bewusst und eindeutig erteilt werden, § 13 Abs. 2 Nr. 1 TMG. Dies setzt voraus, dass der Betroffene über Art, Umfang und Zweck der Datenverarbeitung sowie deren Stattfinden in einem Drittstaat unterrichtet wird. Außerdem genießen Daten, die Rückschlüsse auf den Gesundheitszustand einer Person zulassen, einen besonderen Schutz. In die Verarbeitung sensibler Daten kann nur durch ausdrückliche Erklärung eingewilligt werden, §§ 4a, Abs. 3, 3 Abs. 9 BDSG. Erforderlich ist mithin eine aktive Handlung des Nutzers, z.B. das Setzen eines Häkchens in einer Checkbox[5]. Hier kann die sog. Zweiklick-Lösung einen datenschutzkonformen Ausweg bieten[6]. Diese ist für die Einbettung von Inhalten, die der User selbstständig abruft, etwa YouTube-Videos, geeignet. Aber auch hier muss der Webseitenbetreiber seinen Informationspflichten gerecht werden. I.d.R. wird dieser jedoch nicht über die erforderlichen Kenntnisse der Datenverarbeitung durch die Drittpartei verfügen. Jedenfalls genügt der bloße Hinweis auf bzw. ein Link zu den Datenschutzerklärungen von Drittparteien nicht aus, um bestehende Informationspflichten zu erfüllen.

Für die Einbettung von Inhalten, die geladen werden und personenbezogene Daten an die Drittpartei weiterleiten, bevor dem Nutzer die Gelegenheit geboten wird, einzuwilligen bzw. bevor der Diensteanbieter seinen Informationspflichten nachkommen kann, ist die Zweiklick-Lösung hingegen nicht praktikabel: Eine Einwilligung muss der Verarbeitung vorausgehen[7]. Daraus ergibt sich, dass selbst die Einwilligung des von der Datenverarbeitung Betroffenen die Einbettung von Fremdinhalten nur in bestimmten Konstellationen legitimieren kann.

1 Die Europarechtskonformität dieser Bestimmung ist Gegenstand des in Fn. 6 zitierten Vorabentscheidungsverfahrens.
2 Anders Voigt/Alich, NJW 2011, 3541-3544 (3542); im Ergebnis krit. gegenüber der Inanspruchnahme von Seitenbetreibern Plitz, CR 2011, 657-664 (664).
3 LG Düsseldorf, MMR 2016, 328-331 (330).
4 IP-Anonymisierung in Analytics, https://support.google.com/analytics/answer/2763052?hl=de, abgerufen am 25.05.2016.
5 Dazu Fröhlisch/Pilous, MMR 2015, 613-636 (634).
6 Dazu Fröhlisch/Pilous, MMR 2015, 613-636 (635).
7 Simitis, wie Fn. 5, § 4a Rn. 27.

Ende des Zitats von Dr. Laura Schulte.

Ausblick

An dieser Stelle ist zu beachten, dass sich die rechtliche Situation zwischenzeitlich wieder verändert hat. Insbesondere die eingeführte Datenschutzgrundverordnung (DSGVO) hat weitreichende Änderungen an der gesetzlichen Lage bewirkt. Es muss zukünftig ebenfalls eine Entscheidung des OLG Düsseldorf zur Haftung bei Verwendung von Like-Buttons berücksichtigt[28] werden. Im Ausblick steht eine ePrivacy-Verordnung, welche den Schutz der elektronischen Kommunikation sicherstellen soll.

8.2.6 *Implementierung*

Umsetzung

In Studie A 8.1.6 wurden Tracker anhand der generierten Netzwerkverbindungen identifiziert. Eine genauere Betrachtung war nicht erforderlich, da die Prüfung der Datenschutzerklärung im Vordergrund stand. Das Messwerkzeug musste dabei die Anforderung erfüllen, verlässlich eine Liste der erzeugten Verbindungen bei Abruf der Hauptseite zu erstellen.

Browser

In dieser Studie ist das Verhalten des Browsers von entscheidender Bedeutung. Welche HTTP-Felder vom Browser bei Abruf einer Ressource übermittelt werden kann je nach Browsertyp abweichen. Ein Beispiel ist der HTTP-Header `Public-Key-Pins` nach RFC 7469 [51], der nicht von jedem Browser unterstützt[29] wird.

Ziel

Je stärker sich Analysen den Clients annähern, die auch ein üblicher Internetnutzer verwendet, desto präziser können diese durchgeführt werden. Ein praktisches Ziel dieser Studie ist deshalb, ein Werkzeug zu entwickeln, welches eine möglichst realitätsnahe Untersuchung durchführen kann. Anschließend können die erhobenen Daten auf unterschiedliche Weise ausgewertet und interpretiert werden.

Überblick

In diesem Abschnitt wird das Werkzeugs nur grob beschrieben, weil es im dritten Teil der Dissertation ab Kapitel 9 eine zentrale Rolle einnimmt.

28 OLG Düsseldorf: Zur Haftung und Verbandsklagebefugnis bei Verwendung des Like-Buttons - Kommentar von Laura Schulte in: Kommunikation und Recht 2017, S. 198-199
29 https://web.archive.org/web/20170702171419/https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Public-Key-Pins, abgerufen am 15.01.2018.

Werkzeugformen

In Abschnitt 5.2.1 wurden verschiedene Werkzeugformen aus verwandten Arbeiten vorgestellt. Unter Berücksichtigung der besonderen Anforderungen, die im dritten Teil dieser Dissertation an das Werkzeug gestellt werden, schließt die Nutzung bestehender Implementierungen aus.

Browserautomatisierung

Die hier gewählte Werkzeugform ist die Browserautomatisierung. Dabei wird ein gewöhnlicher Browser eingesetzt, wie er üblicherweise von Internetnutzern verwendet wird. Die bestehende Implementierung setzt auf den Firefox 45, welches über Selenium 2.52.0 in der Programmiersprache Python (2.7.11) angesteuert wird. Zum Zeitpunkt der Studie galten diese Anwendungen als aktuell.

Datenmessung

Eine technisch kritische Fragestellung ist die Extraktion der gewonnenen Daten. Bestehende Implementierungen, wie z. B. OpenWPM[30] [48], verwenden Browsererweiterungen (Plugins) zum Aufzeichnen der Daten. Um auf die Nutzung solcher Browsermodifikationen zu verzichten, wurde eine eigene Analysemethode entwickelt.

HAR-Log

Seit Mozilla Firefox Version 41 unterstützt selbiger den Export eines HAR-Logs (HTTP Archive[31]). Ein solches Log listet die Netzwerkkommunikationen auf, die der Browser mit Abruf einer Webseite tätigt und ermöglicht das Speichern in einem maschinenlesbaren Format (JSON). In Quelltext 8.1 ist ein solcher HAR-Log-Eintrag beispielhaft abgebildet. Zu sehen ist ein GET-Request zu einer Klinikhauptseite übertragenen HTTP-Header.

```
1
2   "entries": [
3     {
4       "pageref": "page_1",
5       "startedDateTime": "2016-04-17T16:52:25.103+02:00",
6       "time": 41,
7       "request": {
8         "bodySize": 0,
9         "method": "GET",
10        "url": "http://kemperhof.gk.de/startseite/index.html",
11        "httpVersion": "HTTP/1.1",
12        "headers": [
13          {
14            "name": "Host",
15            "value": "kemperhof.gk.de"
16          },
17          {
18            "name": "User-Agent",
19            "value": "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101
                  Firefox/45.0"
20          },
21          {
22            "name": "Accept",
23            "value": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
24          },
25          {
26            "name": "Accept-Language",
27            "value": "de,en-US;q=0.7,en;q=0.3"
28          },
29          {
30            "name": "Accept-Encoding",
31            "value": "gzip, deflate"
32          },
33          {
34            "name": "Connection",
35            "value": "keep-alive"
36          }
37        ],
38        "cookies": [],
```

30 https://github.com/citp/OpenWPM, abgerufen am 15.01.2018.

31 http://www.softwareishard.com/blog/har-12-spec/, abgerufen am 15.01.2018.

```
39          "queryString": [],
40          "postData": {
41            "mimeType": "",
42            "params": [],
43            "text": ""
44          },
45          "headersSize": 320
46        },
47        "response": {...}
```

Quelltext 8.1: Beispiel eines HAR-Log-Eintrags.

Ablauf

Die Automatisierung umfasst die folgenden Schritte:

- Übergabe einer Internetadresse (URI), die analysiert werden soll.
- Starten des Browsers und Abruf der angegebenen Adresse.
- Nach Abschluss des Ladevorgangs wird das HAR-Log im Profilordner abgelegt.
- Das HAR-Log wird extrahiert und der Browser geschlossen.
- Die Ergebnisdatei wird bei Bedarf in eine Datenbank übertragen.

Analyse der Daten

Verarbeitung

Die extrahierten Daten werden in eine HAR-Datei geschrieben. Diese wurde zum Zwecke der Analyse in eine Datenbank überführt. Im Anhang A.4 wird das dabei verwendete Datenbanklayout abgebildet.

Datenanalyse

Durch eine Analyse der Anfragen (Requests) und deren Antworten (Responses) lassen sich die Drittparteiaufrufe erheben, die durch eine Klinikwebseite verursacht wurden, wie in Quelltext 8.2 zu sehen ist.

```
1  SELECT request.id,
2         requestid,
3         request.url,
4         responseid,
5         starteddatetime
6  FROM   entries
7         LEFT JOIN request
8                ON ( request.id = entries.requestid )
9  WHERE  entries.instcheckid = INST
```

Quelltext 8.2: Erhebung der HTTP-Requests und -Responses; *INST* steht für die jeweilige Klinik.

Analysemethoden

Es wurden die Analysen der am häufigsten eingebetteten Drittparteien, die Abdeckungsanalyse und ein Netzwerkgraph durchgeführt. Wie solche Analysen umgesetzt werden, wurde bereits in Abschnitt 6.1.3 im ersten Teil der vorliegenden Arbeit beschrieben. Aufgrund der geringeren Anzahl an Knoten ist es möglich, die Knoten und Kanten deutlicher zu visualisieren.

8.2.7 *Ausführung*

Testmenge

Eine Datenquelle mit einer Auflistung aller Webseiten von Krankenhäusern und Kliniken der Bundesrepublik Deutschland konnte nicht gefunden werden. Es zeigt sich, dass dies in die Zuständigkeit der Bundesländer fällt. Aus

diesem Grund werden die Online-Datenbanken Kliniken.de[32] und weisse-liste.de (letzteres seit 2017 bekannt als Deutsches Krankenhaus Verzeichnis [33]) durchsucht. Dabei werden entsprechend dem Szenario nur Einrichtungen entnommen, die Lungenkrebs behandeln.

Auswahl

Das Ergebnis sind 1 260 Web-Adressen, deren Anzahl sich durch Entfernung der doppelten Einträge bzw. durch nicht mehr erreichbaren Webseiten auf 835 reduzieren ließen. Die vollständige Liste befindet sich im Anhang D.3.

Ausführung

Zeitpunkt der Erhebung ist der 17.04.2016 um 14:45 Uhr. Während der Analyse sind keine Fehler aufgetreten.

8.2.8 *Ergebnisse*

Verbindungen

Während der Untersuchung wurden insgesamt 41 138 Verbindungen erzeugt, die sich auf 30 237 Erst- und 10 901 Drittparteien aufteilen. Es wurden dabei 1 253 unterschiedliche Drittparteien registriert. Die am häufigsten eingebundenen Drittparteien können in Tabelle 8.5 eingesehen werden.

Referrer/Origin

Bei 39 890 Anfragen wurde das Referrer-Feld gesetzt (Differenz von 1 248). Bei 38 163 Anfragen wurde die besuchte Krankenhauswebseite in diesem übermittelt. In den verbleibenden 1 727 Fällen fand eine weitere Einbettung beim Embedder statt, so dass der Referrer ersetzt wurde. Das Origin-Feld wurde bei 1 011 Anfragen bzw. 147 Einrichtungen übertragen und enthielt die Adresse der Klinik. Sofern eine Drittpartei eingebunden war, wurde in allen Fällen die besuchte Webseite übermittelt. Es zeigt sich, dass Referrer und Origin feste Bestandteile der HTTP-Kommunikation sind und Anbieter mit einer Übermittlung dieser Daten rechnen können.

Google-Analytics

Wie bei der historischen Entwicklung zeigt sich auch hier, dass Google-Analytics eine dominante Position einnimmt: 38 % (317) der Webseiten setzen diesen Dienst ein. Ganz besonders stark zeigt sich der Einsatz von Google-Diensten aus vermeintlich funktionalen Gründen: Schriftarten (fonts.gstatic.com, fonts.googleapis.com), Programmbibliotheken, Kartendienste, statische Inhalte und Videos. Diese sind ebenfalls auf den Hauptseiten eingebettet und stammen vom gleichen Anbieter.

Abdeckung

Während von Google-Analytics nur ein Drittel abgedeckt wird, zeigt die Abdeckungsanalyse in Abbildung 8.3, dass das Unternehmen Google durch seine zusätzlichen Angebote die Reichweite auf fast 2/3 (61 %) erhöht. Auch wenn Webseiten solcher Einrichtungen nicht direkt mit Gesundheitsdaten der Besucher in Berührung kommen, kann Google Besuchsinformationen bei 2 von 3 Krankenhauswebseiten erheben.

Graph

Der in Abbildung 8.4 dargestellte Graph zeigt die Erstparteien (schwarze Knoten) und Drittparteien (rote Knoten). Wird auf der Startseite einer Klinik die Einbettung zu einer Drittpartei registriert, wird dies durch eine Kante dargestellt.

32 https://www.kliniken.de/, abgerufen am 07.07.2017.

33 http://www.deutsches-krankenhaus-verzeichnis.de/, abgerufen am 07.07.2017.

#	Drittpartei	Kategorie	Anzahl (Anteil)
1	www.google-analytics.com	Web Analytics	317 (38.0%)
2	fonts.gstatic.com	Fonts	189 (22.6%)
3	fonts.googleapis.com	Fonts	184 (22.0%)
4	ajax.googleapis.com	Bibliothek	128 (15.3%)
5	www.google.com	Suchmaschine	68 (8.1%)
6	maps.google.com	Kartendienst	52 (6.2%)
7	csi.gstatic.com	statische Inhalte	44 (5.3%)
8	s.ytimg.com	Videoportal	44 (5.3%)
9	i.ytimg.com	Videoportal	41 (4.9%)
10	stats.g.doubleclick.net	Web Analytics	41 (4.9%)
11	www.youtube.com	Videoportal	40 (4.8%)
12	maps.googleapis.com	Kartendienst	34 (4.1%)
13	static.doubleclick.net	Web Analytics	34 (4.1%)
14	piwik.sana.aspr.de	Web Analytics	30 (3.6%)
15	www.facebook.com	Soziales Netzwerk	27 (3.2%)

Tabelle 8.5: Übersicht der am häufigsten eingebetteten Drittparteien.

8.2.9 *Evaluation*

Überblick

Mögliche Vorgehensweisen zur Evaluierung dieser Studie sind:

- Ein Vergleich der Ergebnisse mit verwandten Arbeiten und
- die Prüfung der Funktionsfähigkeit des verwendeten Werkzeugs.

Vergleich

Für einen Vergleich der Studienergebnisse fehlt es an verwandten Arbeiten. Abweichungen mit anderen quantitativen Studien wie z. B. mit Teil 1 lassen sich durch die speziell gewählte Zielgruppe (Webseiten im Gesundheitswesen) erklären.

Werkzeugevaluation

Aus diesem Grund kann nur eine Evaluierung des verwendeten Werkzeugs durchgeführt werden. Wie bei der Implementierung bereits beschrieben wurde, ist dieses Werkzeug eine zentrale Komponente im dritten Teil dieser Arbeit. Die HAR-log basierte Analyse, der WebChecker, wird in Kapitel 10 in Abschnitt 10.5.1 ausführlich beschrieben und anschließend in Abschnitt 11.2.2 evaluiert. Ein manueller Vergleich der Messergebnisse mit denen eines anderen Browsers belegen die Effektivität dieser Analyseform.

8.2.10 *Diskussion*

Forschungsfrage

Im Folgenden soll zunächst die in Abschnitt 8.2.3 gestellte Forschungsfrage beantwortet werden:

> **In welchem Ausmaß finden Einbettungen von Drittinhalten auf Webseiten im Gesundheitswesen statt und was sind die Konsequenzen?**

Monopolbildung

Die technische Analyse hat gezeigt, dass Einbettungen auf vielfachem Weg die Verfolgung bzw. das Tracken eines Nutzers ermöglichen. Allen

voran ist die IP-Adresse als personenbezogenes Merkmal und der HTTP-Referrer als Informationsquelle für die besuchte Webseite. Darüber hinaus ergeben sich quer durch die Protokollschichten weitere Möglichkeiten zur Identifizierung des Nutzers. Ein Anbieter allein ist in der Lage, 2 von 3 Krankenhaus- bzw. Klinikwebseiten zu überwachen. Dies unterstreicht die Monopolbildung, wie sie bereits im ersten Teil dieser Arbeit beschrieben wurde.

Rechtliche Betrachtung

Die rechtliche Analyse zeigte, dass die durch Einbettungen bewirkte Datenweitergabe eine Erwähnung in der Datenschutzerklärung rechtfertigt bzw. sogar eine Einwilligung durch den Besucher voraussetzt. Es ist jedoch damit zu rechnen, dass eine solche Forderung viele Dienstanbieter überfordern würde.

Verstecktes Tracking

Als Tracking durch die Hintertür ist die Situation zu bezeichnen, dass eine Nutzerverfolgung über funktionale Bestandteile einer Webseite durchgeführt werden kann. Dies geht mit der Übermittlung von personenbeziehbare Daten einher. Dabei sticht insbesondere Google (bzw. Alphabet) durch Angebote wie Google Mail, Google+ (Social Media), Google Play (Android), YouTube, etc. hervor.

Kenntnisse

Insbesondere diese Fälle entziehen sich möglicherweise vollständig der Kenntnis des Webseitenbetreibers. Es muss eingesehen werden, dass Einbettungen jeglicher Art zukünftig wie „klassische“ Tracker behandelt werden müssen. Dies umfasst sowohl den sparsamen Einsatz als auch die Möglichkeit für Besucher, diese wahrnehmen und ablehnen zu können.

Gefahr

Beide Seiten, sowohl Nutzer (Verbraucher) als auch Betreiber (Anbieter), müssen stärker für Datenschutzaspekte sensibilisiert werden. Es muss als Aufgabe für Schulen, Berufsbildende Schulen und Hochschulen bzw. Universitäten angesehen werden, ein grundlegendes Gefahrenbewusstsein (Awareness) auf diesem Gebiet zu schaffen.

Weiteres

Ein studienübergreifendes Fazit findet in Kapitel 12 statt.

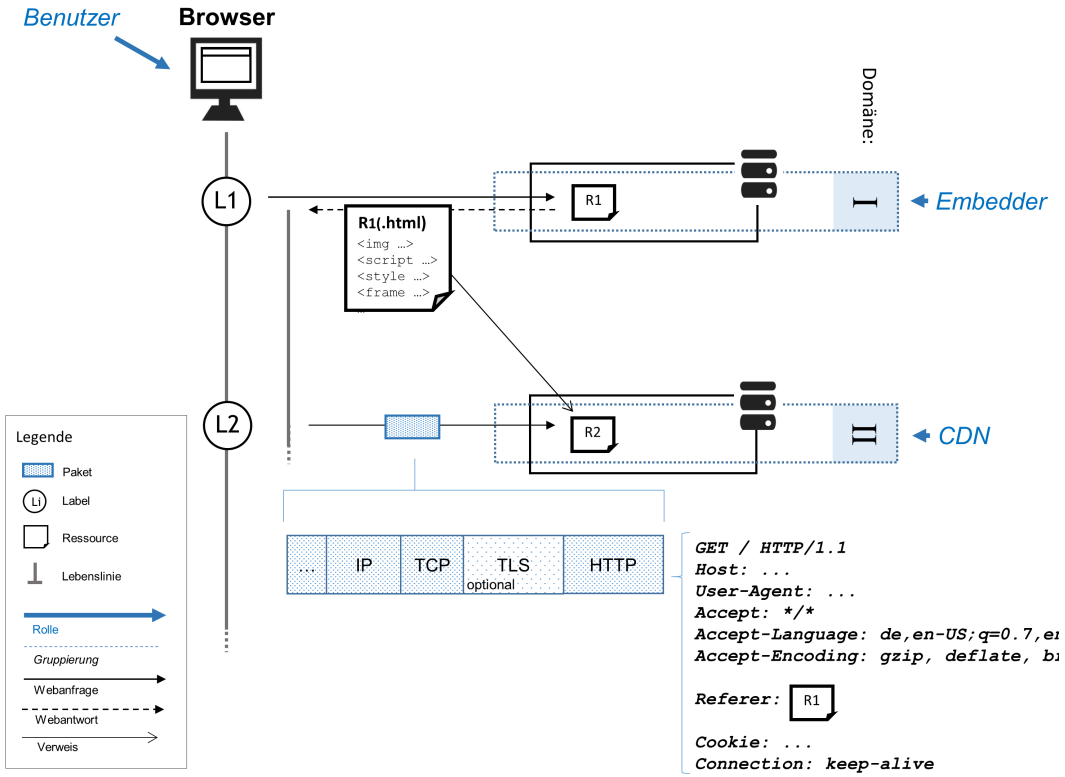

Abbildung 8.2: Weitergabe von Identifizierungsmerkmalen durch Protokollinformationen an den CDN-Betreiber.

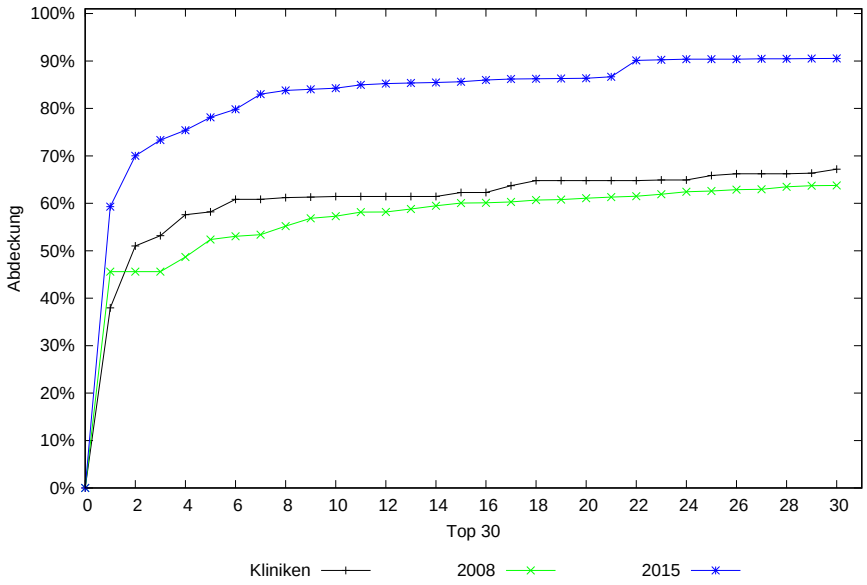

Abbildung 8.3: Abdeckungsanalyse der Top 30 Tracker. Vergleichswerte von 2008 und 2015 stammen aus der retrospektiven Analyse, vgl. Abbildung 6.2.

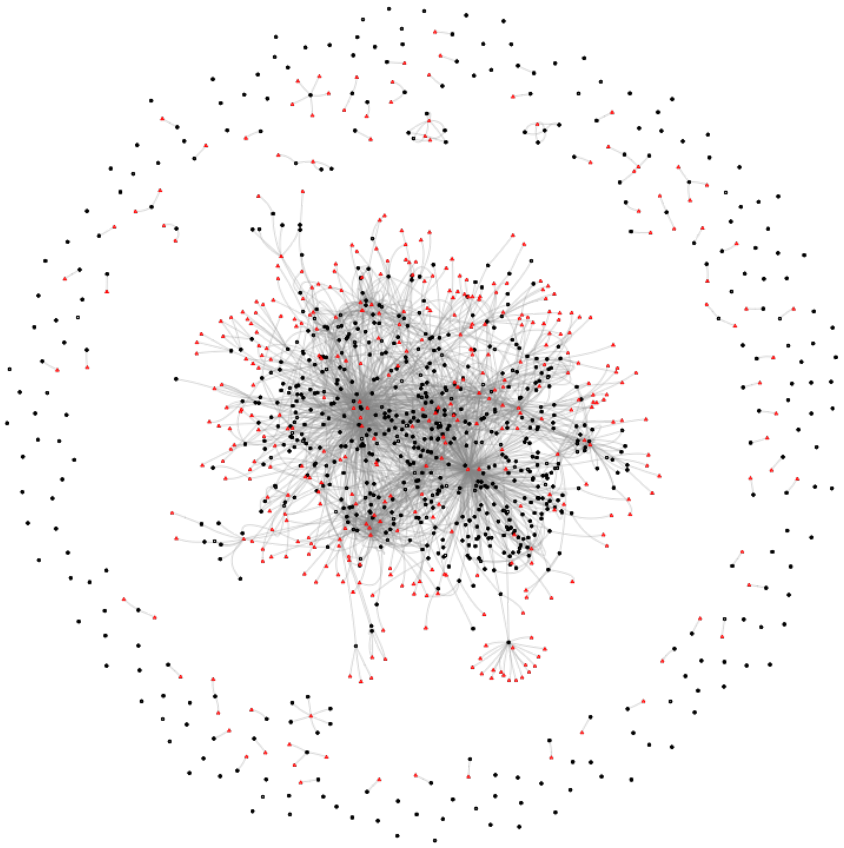

Abbildung 8.4: Graphische Übersicht des Trackingnetzwerkes von Krankenhaus- und Klinikwebseiten (schwarz) bzw. eingebundene Drittparteien (rot).

Teil III

WEB-TRACKING IN DER ZUKUNFT

Nach einer tieferen Betrachtung der Sandbox als Werkzeug wird in diesem Teil der Arbeit mit DisTrack ein neues Analysefra-mework vorgestellt. Diese Form der externen Beobachtung des Browserverhaltens bei Abruf von Webseiten ist ein innovativer Ansatz, um Web-Tracking ohne Nutzung der Browserschnitt-stellen zu erkennen. Ein Gesamtfazit bildet den Abschluss der vorliegenden Arbeit.

9 DIE SANDBOX ALS WERKZEUG

Zusammenfassung: Im Bereich der dynamischen Schadsoftwareanalyse werden Sandbox-Verfahren eingesetzt, um das Verhalten einer Anwendung während der Ausführung zu messen. Im folgenden Kapitel werden Grundlagen zur Funktionsweise von Sandbox-Systemen aufgezeigt und bestehende Lösungen näher betrachtet. Damit wird eine Basis für die Fragestellung geschaffen, ob diese Systeme auch bei der Analyse von Web-Tracking-Verfahren als Unterstützung eingesetzt werden können.

9.1 EINLEITUNG

Dynamische Analysen

Im Bereich der Schadsoftwareanalyse ist die Sandbox ein etabliertes Werkzeug zur dynamischen Analyse von Schadsoftware. Der Begriff Sandbox basiert auf dem Einsatz einer speziell präparierten (üblicherweise virtuellen) Maschine, die nach einer erfolgreichen Analyse über einen Sicherungspunkt in ihren ursprünglichen Zustand zurückgeführt werden kann. Auf diese Weise wird eine dauerhafte Kontamination des Analysesystems ausgeschlossen. In einer solchen Umgebung können ausführbare Dateien und Webseiten gestartet und aufgerufen sowie ihre Verhalten analysiert und bewertet werden. Verhalten bezieht sich in diesem Zusammenhang auf die Interaktion mit dem Betriebssystem und dessen Ressourcen.

Web-Tracking

In Abschnitt 7.3 wurde an verschiedenen Beispielen gezeigt, wie Web-Tracking-Verfahren sich Speicherungen auf dem Nutzersystem verwenden, um eine langfristige Identifizierung durch den Tracker zu ermöglichen. Dies geschieht möglicherweise auch dann, wenn der Nutzer die gespeicherten Daten entfernt hat. Dafür werden legitime und für diesen Zweck vorgesehene Speicherverfahren eingesetzt, sowie Speicherformen, die zwar aus anderen Gründen geschaffen wurden, aber für Tracking ausgenutzt werden können (z. B. der Browsercache). In den beschriebenen Fällen wird stets eine dauerhafte Zustandsänderung am System bewirkt.

Sandbox

Daraus ergibt sich die Frage, ob ein Sandbox-Verfahren zum Aufspüren von Speicherungen genutzt werden kann, die durch Web-Tracking-Verfahren bewirkt werden. Dabei werden Browser und Webseite gemeinsam betrachtet und die Effekte auf das umliegende System gemessen. Die Möglichkeit der Rückführung eines virtuellen Systems in den ursprünglichen Zustand (Snapshot) ermöglicht die Beseitigung dieser Effekte und kann darüber hinaus zu einer Verbesserung der Analyse beitragen. Zu berücksichtigen ist, dass die Sandbox ein generisches Werkzeug ist. So kann jeder Browser oder sonstiges Analysewerkzeug innerhalb dieser Umgebung ausgeführt und anschließend analysiert werden.

Aufbau

Um diese Frage in Kapitel 10 klären zu können, muss die Funktionsweise einer Sandbox verstanden werden. Die hierfür notwendigen technischen Grundlagen werden in Abschnitt 9.2 geschaffen. Anschließend werden bestehende Lösungen in Abschnitt 9.3 betrachtet und der Aufbau eines konkreten Sandboxsystems in Abschnitt 9.4 näher beschrieben. Abschließend wird in Abschnitt 9.5 ein erster Ausblick auf den möglichen Mehrwert einer Sandbox zur Messung von Web-Tracking gegeben.

9.2 Technische Grundlagen

9.2.1 *Virtualisierung*

Definition

Popek und Goldberg [142] definierten bereits im Jahr 1974: „A virtual machine is taken to be an efficient, isolated duplicate of the real machine". Nach Egele et al. [46] ist die Hauptaufgabe von Virtualisierung, den direkten Zugriff auf die Hardware der Maschine zu unterbinden bzw. durch Software zu steuern. Auf diese Weise sind Zustandsänderungen des Systems nicht dauerhafter und physikalischer Natur. Der Unterschied zu einem Emulator besteht darin, dass ihrer Architektur des Hosts und des Gastes identisch sind. Bei Emulation wird der ausgeführten Software eine Hardwareplattform simuliert, die sich in der Architektur von der physikalischen unterscheidet. Allerdings können viele Virtualisierungslösungen zusätzlich auch nicht existierende Teile der Hardware emulieren. Neben der vollen Virtualisierung, die hier näher betrachtet wird, sind auch teilweise, spezielle und anwendungsbezogene Virtualisierungsformen möglich, wie von Greamo und Gosh [64] näher beschrieben werden.

Komponenten

Basis einer Virtualisierung ist der Virtual Machine Monitor (VMM), auch als Hypervisor bekannt. Dieser ermöglicht die Ausführung multipler, voneinander isolierter Systeme (virtuelle Maschinen) auf einer gemeinsamen Hardware. Der Hypervisor schafft auf der virtuellen Maschine die Illusion einer eigenen, exklusiven Hardware. Diese umfasst:

- CPU,
- Arbeitsspeicher,
- persistenter Speicher,
- Netzwerkadapter,
- Grafik- und Soundhardware,
- Tastatur und Maus.

Darüber hinaus ist die Mitnutzung weiterer Anschlüsse des physikalischen Systems (z. B. USB) möglich.

Hosted vs. Bare

Man unterscheidet zwischen zwei Arten von Virtualisierung: `Bare metal` und `Hosted` Virtualisierung. Bei `Bare metal` steuert der Hypervisor direkt die Hardware an, während bei der `Hosted` Variante der Hypervisor als Anwendung in einem Betriebssystem (OS) ausgeführt wird. Sofern es sich um eine `Hosted` Variante handelt, wird unterhalb des Hypervisors ein

Betriebssystem ausgeführt, welches als Hostsystem bezeichnet wird. Der Hypervisor ermöglicht hier als Anwendung die Ausführung von virtuellen Maschinen, der so genannten Gastsysteme. Der Hypervisor regelt unter Berücksichtigung der Beschränkungen des darunterliegenden OS und der Hardware die verfügbaren Ressourcen der Gastsysteme. In der `Bare metal` Variante entfällt das Hostsystem und der Hypervisor tritt an diese Stelle.

Snapshots

Die Generierung von Snapshots ist ein wesentlicher Vorteil von virtuellen Maschinen und übernimmt eine wichtige Funktion zur Analyse von Schadsoftware in virtuellen Umgebungen. Dabei wird vom flüchtigen und persistenten Speicher ein Sicherungsabbild erzeugt. Durch ein Abbild kann die Maschine zu einem späteren Zeitpunkt in einen gesicherten Zustand zurückgeführt werden. Nach vollständiger Ausführung einer Schadsoftware in der virtuellen Maschine findet anschließend der Rücksprung zum Zustand vor dem Startzeitpunkt statt. Damit wird eine dauerhafte Kompromittierung des Systems ausgeschlossen wird. Bevor das System zurückgeführt wird, kann ein vollständiges Abbild des Arbeitsspeichers erzeugt und zur Analyse mittels forensischer Werkzeuge (z. B. Volatility[1]) genutzt werden.

Sicherheit

Der Einsatz zur Schadsoftwareanalyse führt zu Fragen bzgl. der Sicherheit von virtuellen Maschinen, insbesondere in Bezug auf die Isolation zwischen Gast- und Hostsystem. Es muss berücksichtigt werden, dass der Hypervisor eine komplexe Softwarekomponente ist, die wie jede andere Software Schwachstellen aufweisen kann. Als Beispiel wurden für den XEN Hypervisor im Jahr 2017 sechs Schwachstellen[2] mit einem CVSS-Score ≥ 9 gemeldet. Besonders kritisch ist CVE-2017-10912 einzuschätzen: „Xen through 4.8.x mishandles page transfer, which allows guest OS users to obtain privileged host OS access, aka XSA-217“. Diese ermöglicht den Übergriff vom Gast- auf das Hostsystem, was wiederum zu einer möglichen Kompromittierung aller Gastsysteme führen kann. Infolgedessen muss stets die Aktualität und Herstellerunterstützung der Virtualisierungssoftware sichergestellt sein. Scarfone et al. [158] geben einen Überblick über die Sicherheit durch Virtualisierung.

Weiteres

Einen vertiefenden Einblick in das Themengebiet der Virtualisierung bieten Portnoy [143] und Blokdijk [22].

9.2.2 *Windows API und Systemaufrufe*

Betriebssystem

Nach Tanenbaum [179] verfolgen Betriebssystem zwei Aufgaben: Abstraktionen für Benutzerprogramme zur Verfügung zu stellen und die Betriebsmittel des Computers zu verwalten. Zum Zwecke dieser Abstraktion stellt das Betriebssystem fertige Programmbibliotheken bereit, welche in eigene Applikationen integriert werden können. Genutzt werden diese zur

- Prozessverwaltung,

1 http://www.volatilityfoundation.org, abgerufen am 11.02.2018.

2 https://www.cvedetails.com/vulnerability-list/vendor_id-6276/XEN.html, abgerufen am 10.02.2018.

- Datei- und Verzeichnisverwaltung,
- Generierung von Benutzeroberflächen oder sonstigen grafischen Ausgaben,
- Steuerung von externen Geräten,
- Zugriffnahme von Netzwerkressourcen und viele weitere.

Diese können statisch in die Anwendung integriert werden (static linking) oder, was häufiger[3] der Fall ist, dynamisch eingebunden und aus der Anwendung aufgerufen werden (dynamic oder runtime linking).

Windows APIs

Auf Windowssystemen sind häufig die Bibliotheken `User32.dll`, `Kernel32.dll`, `Advapi32.dll` genutzt. Selbsterklärende Beispiele sind die folgenden Funktionen der `Kernel32.dll`: `CreateDirectoryW`, `DeleteFileW`, `OpenProcess`, `ReadFile`, `WriteFile`. Diese Funktionen werden unter den Begriff Windows Application Programming Interface (API) zusammengefasst. Eine Übersicht gibt der Windows API Index[4]. Zusätzlich zu den APIs finden sich native Funktionen in der `ntdll.dll`. Während die Windows APIs über verschiedene Windowsversionen hinweg in ihrer Signatur[5] unverändert bleiben, können sich in nativen Funktionen Änderungen ergeben. Aus diesem Grund wird die Nutzung der API anstelle nativer Funktionen empfohlen.

Berechtigung

Während einige dieser Funktionen nur einfache Rechte auf dem System benötigen (User-Mode), gelten andere als privilegiert (Kernel-Mode). Üblicherweise werden Anwendungen nur mit einfachen Berechtigungen ausgeführt, während die privilegierten dem Betriebssystem vorbehalten sind. Dies ist vor allem zur Koordination von Systemressourcen notwendig, die einen exklusiven Zugriff voraussetzen. Ein Beispiel ist das Lesen und Schreiben von Dateien auf einem Datenträger: Eine solche E/A (bzw. I/O) Operation ist nur im Kernel-Mode zulässig. Zu diesem Zweck stehen Systemaufrufe (system calls) zur Verfügung, die vom User-Mode in den Kernel-Mode schalten und dem Betriebssystem die Ausführung der gewünschten Funktion im Kernel-Mode überlasst. Anschließend wird wieder in den User-Mode zurückgeschaltet und das Ergebnis der Ausführung kann durch die Anwendung verarbeitet werden. Bei einem Systemaufruf handelt es sich demnach um einen Aufruf einer Betriebssystemfunktion, die nur im Kernel-Mode ausgeführt werden kann.

Weiteres

Eine detaillierte Übersicht der internen Abläufe bei Aufruf von Systemfunktionen im Windows Betriebssystem geben Russinovich et al. [155, S. 132 ff.]. Weitere Informationen zur Windows API ist bei Rector u. Newcomer [149] und Simon [167] zu finden. Sikorski und Honig geben eine Übersicht genutzter APIs von Keyloggern [165, S. 19].

3 So Sikorski und Honig: „Of all linking methods, dynamic linking is the most common and the most interesting for malware analysts.“, Quelle: [165, S. 16].

4 https://msdn.microsoft.com/de-de/library/windows/desktop/ff818516(v=vs.85).aspx, abgerufen am 10.02.2018.

5 Gemeint ist die Definition der Schnittstelle: dies umfasst Name, Parameter und Rückgabewert sowie deren Datentypen.

9.2.3 *Schadsoftwareanalyse*

Statisch vs. Dynamisch

Zur Analyse von Schadsoftware kann grundsätzlich zwischen zwei Verfahrensweisen gewählt werden: die statische und die dynamische. Zur statischen Analyse von Schadsoftware werden nur die beinhaltenden Daten des analysierenden Objektes betrachtet. Bei ausführbaren Programmen (.exe) kann sich dies bis zur zeilenweisen Analyse des Maschinencodes erstrecken. Bei einer dynamischen Analyse wird die Anwendung in einer speziell präparierten Umgebung ausgeführt wobei die Seiteneffekte auf das umliegende System gemessen werden.

Statische Analyse

Umsetzung

Bei einer statischen Analyse [165, 33] wird eine Anwendung betrachtet, ohne sie im Prozessor auszuführen. Es handelt sich um ein Reverse Engineering: die Extraktion der Verhaltensweise aus der fertigen Anwendung. Der Maschinencode wird toolgestützt sowie schrittweise und systematisch analysiert. Aus Gründen der Effizienz werden bei der Analyse bestimmte Bereiche der Anwendung genauer fokussiert. Dazu zählen die verwendeten Bibliotheken, die in der Import-Tabelle (Import Table) der ausführbaren Anwendung angegeben werden. Wird beispielsweise die `WSock32.dll` oder `Wininet.dll` eingebunden, lässt dies eine Netzwerk- bzw. Internetfähigkeit vermuten. Ferner können auch die aus diesen Bibliotheken importierten Funktionen berücksichtigt werden, wie sie im Abschnitt 9.2.2 bereits erläutert wurden. Der Systemaufruf „RegisterHotKey“[6] der Bibliothek `User32.dll` gilt beispielsweise als ein üblicher Aufruf von KeyLoggern, der deshalb im Besonderen in der Überwachung von Antivirensoftware steht. Schreibende Zugriffe auf bestimmte Orte eines Betriebssystems können zu automatischen Startvorgängen von Anwendungen führen; gespeicherte Zeichenketten (Strings) können E-Mail-Adressen oder URIs beinhalten.

Fingerprint

Auch ein Fingerprint, die Erstellung eines Hashwertes über ein kryptografisch sicheres Hashverfahren, kann als Teil einer statischen Analyse angesehen werden. Zwar offenbart diese selbst keine weiteren Informationen, dient jedoch zum Auffinden bereits durchgeführter Analysen in einschlägigen Datenbanken. Es muss bedacht werden, dass nur durch eine einzelne Bitänderung, möglicherweise das Datum der Kompilierung, bereits zu einem neuen ggf. unbekannten Hashwert führt.

Weiteres

Shankarapani et al. [163] geben eine Übersicht über Probleme und möglichen Lösungen bei statischen Analysemethoden. Einen Einstieg in die softwaregestützte statische Analyse von Schadsoftware bieten Sikorski und Honig [165].

6 https://msdn.microsoft.com/de-de/library/windows/desktop/ms646309(v=vs.85).aspx, abgerufen am 12.02.2018.

Dynamische Analyse

Problemstellung

Eine rein statische Analyse kann bereits in einem sehr frühen Stadium der Analyse misslingen, z. B. bei gepackten Executables. Gemeint ist damit die Integration der Dekompression als Teil des Programmcodes. Dabei wird die gesamte Anwendung als Datenpaket komprimiert und die Dekompressionsfunktion als ausführbarer Code hinzugefügt. Dies sorgt für die selbstständige Dekompression bei Ausführung der Anwendung. Ein bekanntes Verfahren ist UPX (Ultimate Packer for executables), welches für 32-Bit und 64-Bit Anwendungen verwendet werden kann. Bei einer statischen Analyse würden nur die eingebundenen API-Aufrufe sichtbar, welche die Dekompressionsfunktion benötigt. Alle weiteren wären in der komprimierten Zeichenkette versteckt und für eine statische Analyse zunächst unsichtbar. Um zum Kern der Anwendung vorzudringen, muss diese Kompressionsschicht zunächst entfernt werden. Dabei muss berücksichtigt werden, dass der Entwickler der Schadsoftware nicht auf standardisierte Verfahren beschränkt ist. Vielmehr ist dieser in der Lage, der Anwendung eigene Kompressions- und Verschlüsselungsschichten hinzuzufügen.

Lösung

Eine solche manuelle Freilegung ist mit einem hohen Aufwand verbunden, dem ein hohes Aufkommen von Schadsoftware gegenübersteht[7]. Aus diesem Grund werden dynamische Analyseverfahren eingesetzt, die vergleichsweise weniger Zeit und Aufwand erfordern. Dabei wird in einer speziell dafür vorgesehenen Umgebung die Applikation ausgeführt. Ziel einer solchen Analyse ist es, die wesentlichen Eigenschaften einer Schadsoftware anhand ihres Verhaltens zu erkennen.

Umsetzung

Im Fokus stehen dabei Interaktionen mit dem umliegenden Betriebssystem, insbesondere Systemaufrufe, wie sie in Abschnitt 9.2.2 beschrieben wurden. Durch Prozessüberwachung können diese aufgezeichnet und anschließend analysiert werden.

9.2.4 *Prozessüberwachung*

Verhaltensanalyse

Kern der dynamischen Schadsoftwareanalyse ist das Überwachen des Verhaltens einer ausgeführten Applikation. Die Analyseumgebung muss in der Lage sein, jede Interaktion der Software mit dem umliegenden System zu registrieren und ggf. zu filtern. Sofern eine virtuelle Umgebung zum Einsatz kommt, die eine Wiederherstellung des vorherigen Zustands mittels Snapshot ermöglicht, kann auf eine Filterung verzichtet werden.

Hooking

Diese Überwachung wird durch so genanntes Hooking umgesetzt. Bei diesem Vorgang werden Funktionsaufrufe abgefangen und so verändert, dass nach Aufruf und Abschluss der Funktion eigener Code ausgeführt wird. Dieser kann beliebig ausgestaltet sein, zu Analysezwecken steht jedoch die Aufzeichnung der Funktionsabfolgen im Vordergrund. Hooking erfordert die

7 So auch Willems et al.: „In the face of such threats, security researchers can't combat malicious software using manual methods of disassembly or reverse engineering.“, Quelle: [199, S. 33].

Anpassung der überwachten Funktionen einer ausführbaren Anwendung. Mit DLL-Injection wird initial eine Programmbibliothek in die Anwendung eingefügt und zur eigenen Ausführung gebracht, wodurch dieser Umbau intern in die Wege geleitet wird. Der Ablauf ist daher, mit DLL-Injection eigenen Code in die Anwendung zu bringen, der anschließend für die notwendigen Modifikationen zum Hooking der Funktionsaufrufe sorgt.

Injection

Für DLL-Injection und Hooking stehen je nach Anwendungsfall und Betriebssystem verschiedene Umsetzungsformen zur Verfügung. Bremer [27] stellt umfassende Informationen zu Hooking auf x86-Systemen bereit. Die Maßnahmen sind auf 64-Bit Systemen analog durchführbar[8]. In diesem Abschnitt wird die Verfahrensweise erläutert, die zur Analyse von Schadsoftware in einer Sandbox eingesetzt wird. Weitere Formen wie z. B. kernelseitige Änderungen[9] werden von Hoglund und Butler [79] genauer beschrieben.

DLL-Injection

Übersicht

Bei DLL-Injection [80, 8] handelt es sich um einen Vorgang, in dem eigener Code in eine laufende Anwendung eingefügt wird. Grundsätzlich kommt auch eine Manipulation der Software vor deren Ausführung in Betracht. Es muss beachtet werden, dass eine laufende Anwendung weitere zu berücksichtigende Prozesse starten kann. Aus diesem Grund muss die Übernahme einer laufenden Anwendung möglich sein. Im Folgenden wird eine Zielanwendung oder ein Zielprozess durch eine Quellanwendung (DLL) erweitert, welche anschließend die internen Anpassungen zum Hooking bewirkt.

Windows API

Die Windows API ermöglicht die Interaktion mit laufenden Anwendungen, wenn ausreichende Benutzerrechte vorliegen. Die `OpenProcess` Funktion der `Kernel32.dll` liefert im Erfolgsfall ein Handle[10] zurück. Mit diesem Handle können Auskünfte bzgl. der Anwendung (z. B. Anzahl der Threads) erfragt, jedoch auch weitergehende Instruktionen wie das Starten eines neuen Threads innerhalb der Anwendung ausgeführt werden.

Injektion

Nachdem der Prozess geöffnet (attached) ist, muss die Quellanwendung in die Zielanwendung eingefügt werden. Im ersten Schritt wird die Quellanwendung mit `GetFullPathName` gesucht und mit `CreateFileA`[11] gelesen. Anschließend wird mit dem Handle neuer virtueller Speicherbereich in der Zielanwendung alloziert – dazu dient der Aufruf von `VirtualAllocEx`. Im Anschluss daran kann die Quellanwendung mit `WriteProcessMemory` in den Speicher geschrieben werden. Das Ergebnis ist, dass im laufenden Zielprozess die Quellanwendung eingefügt wurde.

8 http://jbremer.org/intercepting-system-calls-on-x86_64-windows/, abgerufen am 12.02.2018.

9 Gemäß den Entwicklern der Cuckoo Sandbox, ein Framework zur Analyse von Schadsoftware, werden aus Gründen der Flexibilität auf kernelseitige Modifikationen verzichtet, vgl. https://github.com/cuckoosandbox/cuckoo/issues/490 (abgerufen am 13.02.2018).

10 Dabei handelt es sich um einen Referenzwert, der in einem bestimmten Kontext die Identifikation der Ressource ermöglicht.

11 CreateFileA dient sowohl der Erstellung neuer als auch der Öffnung zum Lesen bestehender Dateien, vgl. https://msdn.microsoft.com/en-us/library/windows/desktop/aa363858(v=vs.85).aspx (abgerufen am 12.02.2018).

Ausführung

Vorerst ist jedoch nur der Programmcode verfügbar ohne ausgeführt zu werden. Erst durch das Starten eines neuen Threads mit dem Aufruf von `CreateRemoteThread` der Zielanwendung wird die Ausführung bewirkt. Der Thread enthält einen Funktionsaufruf der neu eingefügten Programmbibliothek. Das Betriebssystem ist in der Lage, die Ausführung der Anwendung vor der DLL-Injection zu suspendieren und nach Abschluss weiter auszuführen.

Weitere Techniken

Es ist ein grundsätzliches Problem, dass solche Veränderungen von außen durch die Zielanwendung bemerkt werden können: hier beispielsweise die unerwartete Ausführung eines neuen Threads. Aus diesem Grund werden stetig neue DLL-Injection-Techniken entwickelt. Eine weitere ist die Ausnutzung der APC-Queue[12] (Asynchronous Procedure Calls). Bei dieser Form der asynchronen Programmierung ist jedem Thread eine eigene Warteschlange mit Arbeitsaufträgen zugehörig. Mit der `QueueUserAPC`-Funktion können Aufträge in Form von Funktionsaufrufen in die Queue eingereiht werden. Auf diese Weise kann der Sprung in die zuvor eingefügte Quellanwendung realisiert werden, ohne einen neuen Thread dafür zu starten.

Hooking durch IAT-Modifikation

Überblick

Mit dem Start einer ausführbaren Anwendung werden alle Bibliotheken in den virtuellen Speicher geladen, die im Headerabschnitt *Import Directory Table*[13] (IDT) gelistet sind (.idata-Sektion). Dabei ist der Speicherort, also die Adresse im virtuellen Speicher der Anwendung, nicht konstant, sondern kann aufgrund verschiedener Faktoren abweichen. Für eine Auflösung von Bibliotheken und deren Funktionen zum tatsächlichen Speicherort zur Laufzeit ist die Import Address Table (IAT) zuständig. Wird beispielsweise die `CreateDirectoryA` Funktion der `Kernel32.dll` verwendet, enthält die IAT einen Eintrag, der dieser Funktion eine Adresse im virtuellen Speicher zuweist.

IAT-Modifikation

Eine einfache Form des Hookings ist die Modifikation der *Import Address Table* der Anwendung. Dabei wird die Speicheradresse innerhalb der IAT durch die Adresse einer eigenen Funktion ersetzt. Bei dem oben geschilderten Szenario würde bei einem Lookup nicht die wahre Adresse der `CreateDirectoryA` Funktion zurückgeliefert, sondern die Adresse einer angepassten Funktion mit wahlfreiem Inhalt. Sofern der tatsächliche Speicherort zuvor gesichert wurde, kann anschließend die originale `CreateDirectoryA` Funktion aufgerufen werden, wodurch es zu keinen funktionalen Beeinträchtigungen kommt. Durch das Einbringen einer eigenen Funktion kann eine Überwachung aller Funktionsaufrufe umgesetzt werden.

Nachteile

Diese einfache Form des Hookings besitzt den Nachteil, dass sie nur auf implizit eingebundene Bibliotheken anwendbar ist. So ist es möglich, explizit

12 https://msdn.microsoft.com/en-us/library/windows/desktop/ms681951(v=vs.85).aspx, abgerufen am 12.02.2018

13 https://msdn.microsoft.com/en-us/library/windows/desktop/ms680547(v=vs.85).aspx#import_directory_table, abgerufen am 12.02.2018.

weitere Bibliotheken über den Befehl `LoadLibrary` zur Laufzeit einzubinden und die Adressen der Funktionen durch `GetProcAddress` zu erfragen. In diesem Fall wird die IAT nicht verwendet, wodurch deren Änderungen wirkungslos bleiben. Aus diesem Grund wird Inline Hooking angewendet, wie u. a. von Willems et al. [199] näher beschrieben.

Inline Hooking

Überblick

Die Kernidee, verwendete Funktionen durch neue zu ersetzen, bleibt bei Inline Hooking erhalten. In diesem Fall wird nicht die Adresse der Tabelle, sondern die Funktion selbst angepasst. So werden alle zu hookenden Funktionen durch einen unbedingten Sprungbefehl direkt als erste Anweisung der Funktion erweitert. Grundsätzlich ist der Einschub von neuen Befehlen ohne Änderungen an Adresstabellen nicht möglich. Infolgedessen werden die ersten fünf Bytes, die für den Sprungbefehl benötigt und ersetzt[14] werden, gesichert und in eine neue Trampolin-Funktion überführt. Diese enthält neben den gesicherten Instruktionen auch den Rücksprung zu den verbleibenden Instruktionen hinter den Sprungbefehl der ursprünglichen Funktion. Dies bedeutet, die Originalfunktion wurde durch einen Sprung erweitert und die ersetzten Teile in eine eigene Trampolin-Funktion ausgelagert. Die Trampolin-Funktion enthält die erste(n) Zeile(n) der Originalfunktion sowie den Sprung zu dem unveränderten Teil der Ursprungsfunktion[15]. Aus diesem Grund stellt die Trampolin-Funktion die Funktionalität der ursprünglichen Funktion zur Verfügung und kann als dessen unmodifizierte Version angesehen werden.

Beispiel

Ziel der Modifikation ist, dass die Quellanwendung bekannte und zu überwachende Windows API Funktionen bereitstellt, die in der Zielanwendung ersetzt werden. So steht für jede zu überwachende Windows API eine passende Funktion in der Quellanwendung (DLL) zur Verfügung. Zur Verdeutlichung soll beispielhaft die `CreateDirectoryA`-Funktion der Zielanwendung überwacht werden. Nach dem Hooking führt der Aufruf der `CreateDirectoryA` als erste Anweisung einen Sprung in die injizierte DLL aus. Die DLL sorgt dafür, dass der Zeitpunkt des Aufrufs und die Parameter persistent gesichert werden. Anschließend wird die Trampolin-Funktion aufgerufen, die zur tatsächlichen Ausführung der originalen `CreateDirectoryA` der `Kernel32.dll` führt.

Fortpflanzung

Abschließend muss durch eine spezielle Implementierung von `LoadLibrary` und `LoadLibraryEx` dafür gesorgt werden, dass explizit eingebundene Bibliotheken in gleicher Weise berücksichtigt und modifiziert werden. Auf diese Weise werden alle gewünschten Aufrufe der Windows API überwacht, unabhängig auf welche Weise diese aufgerufen wurden. Somit ist

14 Je nach Anweisung können jedoch nicht blind nur fünf Bytes entfernt bzw. ersetzt werden, da dies längere (bis zu 16 Byte große) Anweisungen zerreißen könnte. In diesem Fall muss mittels einer Length Disassembler Engine die Länge der Anweisungen gemessen und gesammelt werden, bis der Platz für den Sprungbefehl genügt. Der verbleibende Platz kann mit `nop` aufgefüllt werden.

15 Dieser direkte Sprung nach außen gibt der Trampolin-Funktion ihren Namen.

stets eine eindeutige Zuordnung[16] der Aktionen im System zur analysierten Software möglich.

Verbindung mit DLL-Injection

Das Zusammenspiel von DLL-Injection und Inline Hooking wird in Abbildung 9.1 dargestellt. Im ersten Schritt (L1) findet die beschriebene Infiltration des Zielprozesses mit DLL-Injection statt. Dabei wird die `Monitor.dll` in den laufenden `App.exe`-Prozess eingefügt und ausgeführt. Diese Ausführung bewirkt das Inline Hooking (L2) der importierten Funktionen (hier: `CreateDirectoryA`). Dafür wird zur Vereinfachung die erste Anweisung durch einen Sprungbefehl ersetzt. Gesichert wird die ersetzte Operation jedoch in der `CD_Trampoline`-Funktion. Findet nun ein Aufruf der `CreateDirectoryA` aus der `main`-Funktion statt, wird nicht die originale Funktion (importiert aus der `Kernel32.dll`) aufgerufen, sondern die modifizierte `CreateDirectoryA*`. Die erste Anweisung, der unbedingte Sprung (2), führt zur `HookCreateDirectoryA`-Funktion aus der `Monitor.dll`. Diese beinhaltet beliebige Anweisungen: In diesem Fall Instruktionen zur Speicherung des Aufrufzeitpunkts. Anschließend soll die ursprüngliche `CreateDirectoryA` ausgeführt werden, da die main-Funktion entsprechende Rückgaben erwartet. Zu diesem Zweck wird die `CD_Trampoline`-Funktion aufgerufen (3), die zunächst die gesicherte Operation enthält und anschließend den unbedingten Sprung (4) zur zweiten Operation der `CreateDirectoryA*` ausführt. Nach vollständiger Ausführung bzw. vollständigem Abschluss der `CreateDirectoryA*` findet der Rücksprung (5) in die `main`-Funktion statt.

9.3 Sandboxing zur Schadsoftwareanalyse

Überblick

Goldberg et al. [63] stellten 1996 ein modulares Framework zur Untersuchung von nicht vertrauenswürdigen Anwendungen vor. Die Autoren vermuten, dass die Schadwirkung einer Anwendung verringert werden kann, indem der Zugriff auf das darunterliegende Betriebssystem reduziert wird[17]. Zu diesem Zweck werden die Systemaufrufe der ausgeführten Anwendung abgefangen und gefährliche Aktionen gefiltert.

Virtualisierung

Neben der Überwachung von Systemaufrufen wird die Virtualisierung eingesetzt, um eine Kompromittierung des Analysesystems auszuschließen. Aufgrund der Zurückführbarkeit in den Ausgangszustand (Snapshot) ist ein sorgloser Umgang mit dem Gastsystem möglich, wodurch sich die Analogie zu einem Sandkasten (Sandbox) erklärt. Für Obasuyi und Sari [136, S. 5] ist die Sandbox ein wesentlicher Anwendungsfall der Virtualisierung: „Virtual Machine helps in providing secure and isolated environments for applicati-

16 Filterwerkzeuge, wie z. B. der Process Monitor, sind zwar zu einer zeitverzögerten Aufzeichnung von Programmaktivitäten in der Lage, allerdings ist es schwierig, den Aktivitäten einer konkreten Schadsoftware zu folgen. Diese können durchaus unterstützend eingesetzt werden. Umgesetzt wird dies durch ein Filtertreiber, der beim ersten Start in das Betriebssystem installiert wird, vgl. Russinovich et al. [156, S. 413]

17 „An application can do little harm if its access to the underlying operating system is appropriately restricted.“, Quelle: [63, S. 3]

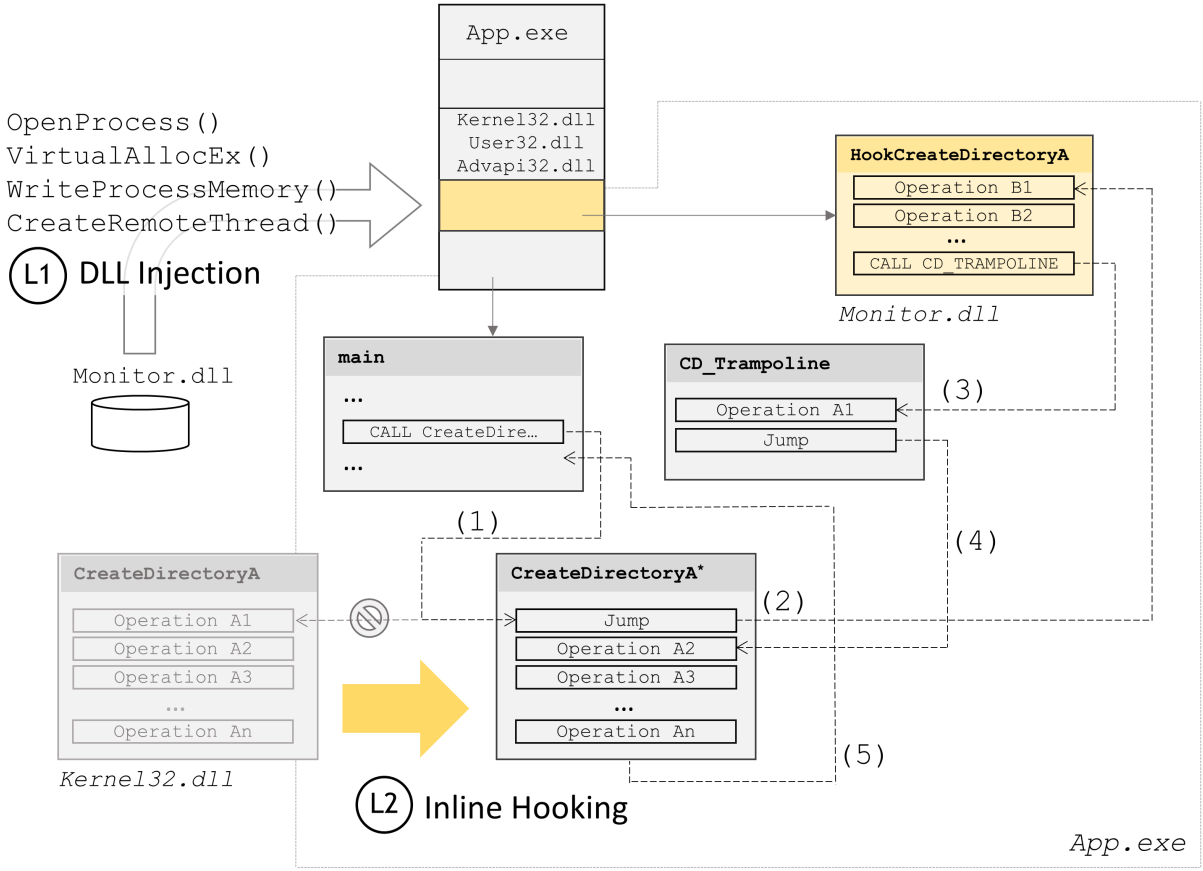

Abbildung 9.1: Visualisierung von DLL-Injection (L1) und Inline Hooking (L2).

ons that are less trusted in the virtualized operating system. Virtualization helps in creating a secure computing environment".

Arten

Es gibt verschiedene Umsetzungen von Sandboxlösungen. Man unterscheidet zwischen Analysewerkzeugen zur Betrachtung von Schadsoftware auf Desktop Rechnern und denen zur Analyse von Apps auf Mobilgeräten. Manche Lösungen bieten allerdings Analysemöglichkeiten für beide Systemarten.

9.3.1 *Sandboxing bei Desktop Applikationen*

Typen

Es muss berücksichtigt werden, dass der Begriff „Sandbox" innerhalb in Informationssicherheit auch in anderen Zusammenhängen verwendet wird. Ein Beispiel ist die Isolation von Java-Anwendungen innerhalb der Java Virtual Machine (JVM). Auch der Google Chrome Browser verwendet eine Sandbox[18], um die mögliche Schadwirkung durch ausgeführte JavaScript-Anwendungen auf das umliegende System zu reduzieren.

Ziele

Auch wenn diese Anwendungsformen zunächst verwandt erscheinen, verfolgen diese unterschiedliche Ziele. Die genannten Beispiele zielen alleine auf eine Isolation von Daten in der Form ab, dass Zugriffe und Modifikationen auf Systemkomponenten unterbunden werden. Bei den hier betrachteten Werkzeugen steht hingegen die Analyse der Anwendung im Vordergrund. Zwar kann eine Isolierung von Daten auch eine Schutzmaßnahme vor Tracking herbeiführen, wie von Pan et al. [139] beschrieben wurde, allerdings bedarf dies der Entwicklung eines eigenen Browsers.

Übersicht

Im Folgenden werden Applikationen und Frameworks betrachtet, die sich zur Sandbox-basierten Analyse von Schadsoftware eignen.

CWSANDBOX Willems et al. [199] präsentieren die CWSandbox zur Analyse von Schadsoftware. Für die Studie wurden 6 148 Schadsoftware-Exemplare mit diesem Werkzeug analysiert und in verschiedene Schadsoftwareklassen eingeteilt. Die Sandbox wurde erst unter dem Namen GFISandbox und später unter ThreatAnalyzer[19] als kommerzielles Produkt weiterentwickelt.

MICS Daisuke et al. [83] stellen mit MicS ein automatisches Analysesystem für Schadsoftware vor, welches die Netzwerkaktivitäten besonders fokussiert. Neben der Aufzeichnung der Netzwerkaktivitäten kommt auch ein Internet-Emulator zum Einsatz, der eingehende Anfragen durch die Schadsoftware entgegennimmt.

ANUBIS Die Anubis Sandbox wurde von dem International Secure Systems Lab entwickelt und dient der Analyse von Anwendungen und Internet-

18 https://chromium.googlesource.com/chromium/src/+/master/docs/design/sandbox.md, abgerufen am 09.03.2018.

19 https://www.threattrack.com/, abgerufen am 19.02.2018.

adressen. Eingesetzt wird sie u. a. von Neugschwandtner et al. [130]. Das Projekt ist mittlerweile aus Zeitmangel eingestellt[20] worden.

CUCKOO Die Sandbox ist aus einem Google Summer Projekt entstanden und wird in Abschnitt 9.4 näher betrachtet. Die Lösung wird in zahlreichen Publikationen eingesetzt [188, 141, 147, 54].

Weiteres

Weitere Werkzeuge aus der Zeit vor 2006, die mittlerweile überwiegend eingestellt wurden, können aus Willems et al. [199, S. 6] entnommen werden.

9.3.2 *Sandboxing bei mobilen Applikationen*

Entwicklung

Insbesondere seit der Einführung des Apple IPhone im Jahr 2007 und der ersten Android basierten Geräte ab 2008 zeigt sich Schadsoftware auch auf Mobilsystemen. In den Jahren von 2011 bis 2016 fand eine stetige Zunahme von schädlichen Apps statt[21]. Die starke Verbreitung von mobilen Geräten sorgt für eine Ausweitung von Schadsoftwaretypen auf diese Systeme[22], die zuvor nur für Desktop-Geräte entwickelt worden sind.

Mobile Sandbox

Um diesen Problemen zu begegnen, wurden Sandboxlösungen für mobile Applikationen geschaffen [23, 173, 186, 38]. Die Mobile Sandbox[23] von Spreitzenbarth et al. [172] ermöglicht eine statische und dynamische Analyse. Während in der statischen Analyse Metainformationen (Manifest) interpretiert werden, findet in der dynamischen Analyse eine Ausführung der Applikation statt, wobei Netzwerkverbindungen und native API-Aufrufe protokolliert werden. Durch das MonkeyRunner-Toolkit[24] werden Benutzeraktionen während der Ausführung der Applikation simuliert.

9.3.3 *Weitere Sandboxlösungen*

Verwandte Lösungen

Im Laufe der Zeit sind Sandbox bzw. Sandbox-ähnliche Lösungen entstanden.

SANDBOXIE. Bei Nutzung von Sandboxie[25] findet eine Anwendungsvirtualisierung statt: eine Insolationsschicht zwischen Betriebssystem und Anwendung. Bei einer mit Sandboxie gestarteten Anwendung werden alle Seiteneffekte am umliegenden System verhindert und zur Applikation hin simuliert. Aus Perspektive der Anwendung reagiert das umliegende System gemäß einer normalen Ausführung. Tatsächlich

20 „Unfortunately, we do not have the resources to maintain these tools and improve them to match an ever-changing malware landscape.“, Quelle: http://anubis.iseclab.org/, abgerufen am 19.02.2018.

21 https://www.gdata.de/blog/2018/02/30489-jede-stunde-rund-343-neue-android-schadprogramme-in-2017, abgerufen am 20.02.2018.

22 https://www.welivesecurity.com/wp-content/uploads/2018/02/Android_Ransomware_From_Android_Defender_to_Doublelocker.pdf, abgerufen am 20.02.2018.

23 https://mobilesandbox.org/, abgerufen am 20.02.2018.

24 https://developer.android.com/studio/test/monkeyrunner/index.html, abgerufen am 20.02.2018.

25 https://www.sandboxie.com/, abgerufen am 20.02.2018.

werden aber alle Änderungen abgefangen und nicht auf das darunterliegende System angewendet. Nach dem Beenden der Anwendung lässt sich dieser Kontext wieder entfernen, wodurch die Ausführung keine Rückstände[26] hinterlässt. Sandboxie wird über DLL-Injection[27] der `SbieDll.dll` und Hooking[28] umgesetzt, wie in Abschnitt 9.2.4 bereits erläutert wurde. Über das Werkzeug Buster Sandbox Analyzer[29] ist eine genauere Auswertung der Programmaktionen möglich. Dieser wurde allerdings seit 2013 nicht mehr weiterentwickelt. Mehr zu Sandboxie findet sich u. a . bei Wojtczuk und Kashyap [202].

DOCKER. Bei Docker[30] handelt es sich um eine verbreitete Virtualisierungslösung, welche Anwendungen und zum Betrieb notwendige Systemteile in gemeinsame Container verkapselt. Grundsätzlich findet dabei eine Trennung der einzelnen Container statt. Docker erleichtert die Herstellung von homogenen Betriebsbedingungen und lässt dabei flexibel Änderungen an selbigen zu. Es muss allerdings als eine „leichtgewichtige" Virtualisierung betrachtet werden, weil Isolationsschichten fehlen, die bei einer vollständigen Virtualisierung üblich sind. Die Sicherheit von Docker wird kontrovers diskutiert und hängt stark von ergänzenden Schutzlösungen wie AppArmor, SELinux oder GRSEC ab[31]. Die Tatsache, dass Docker überwiegend für den Einsatz auf Linux basierten Systemen konzipiert ist, erschwert die Nutzung als Sandbox zur Analyse von Schadsoftware.

Weitere Lösungen wie BufferZone[32] oder Shadow Defender[33] arbeiten analog zu den bereits vorgestellten.

9.4 AUFBAU DER CUCKOO SANDBOX

Übersicht

Nach der Übersicht verfügbarer Sandboxlösungen in Abschnitt 9.3.1 zeigt sich, dass zum aktuellen Zeitpunkt lediglich die Cuckoo zur freien Verwendung ist und einen hinreichenden Reifegrad aufweist, der durch Nutzung in anderen Publikationen belegt wird. Aus diesem Grund wird diese Lösung im folgenden Abschnitt näher betrachtet.

Entstehung

Die Cuckoo Sandbox [36] entspringt einem Google Summer Code Projekt von Claudio Guarnieri aus dem Jahr 2010. Die erste Beta Version war 2011 verfügbar und ist seitdem kontinuierlich bis zur aktuellen Version 2.0.03 weiterentwickelt worden. Im Jahr 2014 wurde die Cuckoo Foundation mit

26 Mit einer genaueren Analyse möglicher Rückstände, die bei einer forensischen Analyse aufgefunden werden können, beschäftigen sich Gupta und Mehte [68].
27 https://www.sandboxie.com/InjectDll, abgerufen am 20.02.2018
28 https://www.sandboxie.com/SBIE_DLL_API, abgerufen am 20.02.2018.
29 http://bsa.isoftware.nl/, abgerufen am 20.02.2018.
30 https://www.docker.com/, abgerufen am 20.02.2018.
31 https://docs.docker.com/engine/security/security/, abgerufen am 20.02.2018.
32 https://bufferzonesecurity.com/, abgerufen am 20.02.2018.
33 http://www.shadowdefender.com/, abgerufen am 20.02.2018.

Sitz in den Niederlanden gegründet, bleibt nach eigener Aussage jedoch frei (im Sinne von Open Source) und ohne Gewinnerzielungsabsichten[34].

Ziele

Die Sandbox ist zur Analyse von

- ausführbaren Windowsanwendungen,
- DLL-Bibliotheken,
- URLs und HTML-Dateien,
- Officedokumenten

und vielen weiteren in der Lage.

Aufbau

Der Aufbau der Sandbox kann sich je nach eingesetzter Virtualisierungslösung unterscheiden. Stets vorhanden ist ein Hostsystem, welches den Analyseprozess steuert und die Auswertung der Messergebnisse vornimmt. Darüber hinaus sind mindestens eins oder mehr Analysesysteme verfügbar, in dem die Schadsoftware zur Ausführung kommt. Für die Übertragung der Ergebnisse ist eine Netzwerkverbindung zwischen Host- und Analysesystem notwendig. Sofern Internetadressen analysiert werden sollen, muss ebenfalls ein Internetzugriff ermöglicht werden. Grundsätzlich werden alle Verbindungen vom Gastsystem nach außen zur Protokollierung über das Hostsystem geleitet, um eine lückenlose Aufzeichnung zu gewährleisten.

Im Folgenden wird ein Überblick über den Aufbau der Sandbox bereitgestellt. Weitere Details können der offiziellen Dokumentation [37] entnommen werden.

9.4.1 *Bestandteile*

Die Sandbox umfasst mehrere Komponenten, die im Folgenden näher betrachtet werden.

SANDBOX. Mit dem Begriff Sandbox wird die gesamte Analyseinfrastruktur adressiert, wie sie nachfolgend näher beschrieben wird.

ANALYSEDATEI. Das *Sample* ist ein Objekt, dass in der Sandbox analysiert werden soll. Dabei kann es sich um eine direkt oder indirekt ausführbare Datei (exe, com, dll) oder um beschreibende Formate handeln, die erst während der Interpretation durch eine verarbeitende Software eine Schadwirkung entwickeln (z. B. pdf oder docx).

GAST. Der Gast ist das Betriebssystem, in dem die Analyse durchgeführt bzw. die Analysedatei ausführt. Hier ist ein Windows XP SP3 oder Windows 7 System empfohlen, weil der überwiegende Bestandteil an Schadsoftware für Windowssysteme entwickelt wird. Das Gastsystem wird virtualisiert, um eine Rückführung in den Zustand vor der Ausführung zu erleichtern. Es ist nicht zwangsweise notwendig: Die Sandbox ermöglicht auch die Nutzung von physikalischen Rechnern, wobei mittels Fog[35] eine Rückführung in den Urzustand umgesetzt wird.

34 https://cuckoosandbox.org/blog/cuckoo-foundation, abgerufen am 13.02.2018.
35 https://fogproject.org/, abgerufen am 13.02.2018.

Im Folgenden wird vom Einsatz einer virtuellen Umgebung ausgegangen.

HOST. Der Host ist das Betriebssystem, in dem der Cuckoo Prozess ausgeführt wird. Als Hostsystem wird ein Linux-basiertes Betriebssystem empfohlen bzw. die Installationsanweisungen sind auf ein solches zugeschnitten. Der Host fungiert ebenfalls als Router und ermöglicht das Routing von IP-Paketen. Auf diese Weise ist sichergestellt, dass alle Netzwerkverbindungen vom Gastsystem ins Internet detektiert werden können.

CUCKOO. Diese Steuerungseinheit befindet sich auf dem Host, nimmt (z. B. über ein Web-Interface) neue Analyseanfragen entgegen, orchestriert parallel ablaufende Analysevorgänge und stellt die Ergebnisberichte zur Verfügung. Ebenfalls steuert dieser Prozess die Virtualisierungssoftware: startet/stoppt Maschinen oder führt den Rollback zu Snapshots aus. Die Konfiguration der gesamten Sandbox wird ebenfalls hier vorgenommen.

AGENT. Der Agent ist ein Prozess, der auf dem Gastsystem gestartet wird. Dieser wartet auf Anweisungen bzgl. neuer Analyseanfragen vom Cuckoo Prozess. Wird ein neuer Analyseprozess eingeleitet, wird der Analyzer (vgl. nächsten Listeneintrag) an den Agent übermittelt und von diesem gestartet.

ANALYZER. Der Analyzer wird vom Cuckoo Prozess an den Agent zur Durchführung einer Analyse übermittelt. Er enthält notwendige Werkzeuge, die für die jeweilige Analysedatei erforderlich sind. Der Cuckoo Prozess ist so in der Lage, für jede Analyse das passende Werkzeug zu wählen und entsprechend anzupassen sowie dynamisch an den Agent zu übertragen. Der Analyzer ist für die detaillierte Überwachung des Prozesses zuständig. Mit den in Abschnitt 9.2.4 beschriebenen Techniken wird die Überwachung technisch umgesetzt.

ANALYSEPAKETE. Die unterstützten Dateiformate erfordern unterschiedliche Analysevorgänge. So müssen beispielsweise Office Dokumente anders behandelt werden als ausführbare Dateien. Oder die Analyse einer Webseite erfordert das Starten des Browsers. So steht für jedes unterstützte Dateiformat ein `Analysis Package` zur Verfügung, welches für die Auswahl der Analyseform verantwortlich ist.

MASCHINENMODULE. Die `Machinary Modules` dienen der Steuerung der Virtualisierungssoftware. Diese müssen je nach Hersteller und Produkt (VirtualBox, KVM, esx, etc.) in unterschiedlicher Weise angesteuert werden. Das Maschinenmodul abstrahiert von den exakten Vorgängen und stellt daher einfache Methoden wie z. B. Start oder Stop zur Verfügung.

HILFSMODULE. Die `Auxilary Modules` werden vor dem Analysevorgang gestartet und nach der Analyse wieder beendet. Sie werden üblicherweise, wenn auch nicht zwingend, auf dem Gastsystem ausgeführt und erfüllen Aufgaben, die nur indirekt für die Analyse erfor-

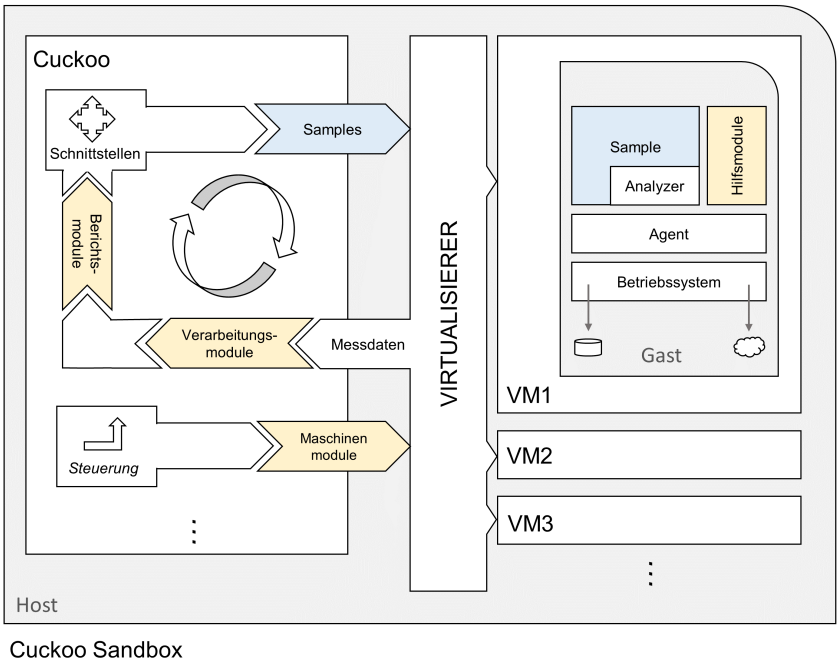

Abbildung 9.2: Bestandteile und Module der Cuckoo Sandbox.

derlich sind. Beispiele sind die Erzeugung von Screenshots auf dem Gastsystem oder die Simulation von Mausbewegungen, um Aktivität durch einen Anwender zu simulieren.

VERARBEITUNGSMODULE. Während der Analyse werden verschiedene Rohdaten gesammelt: Netzwerkmitschnitt, Windows API Aufrufe, Arbeitsspeicherabbild etc. So genannte `Processing Modules` führen anschließend für jede dieser Rohdaten die Aufbereitung in leichter verwertbare Informationen durch. So wird beispielsweise aus der pcap-Datei, also dem Netzwerkmitschnitt, die verschiedenen Verbindungsaufrufe (IP, TCP, DNS, etc.) extrahiert. Dies umfasst sowohl die Daten aus dem Analysepaket als auch die von Hilfsmodulen.

BERICHTSMODULE. Abschließend können durch `Reporting Modules` Verarbeitungsergebnisse weiter aufbereitet, konvertiert oder interpretiert werden. Beispielhaft ist die Überführung der Ergebnisse in eine Datenbank oder die Zusammenfassung in ein PDF Dokument.

TASK. Ein Task besteht aus der Analysedatei oder einer URL, sowie weitere Optionen zur Konfiguration der Umgebung für diesen speziellen Analysevorgang. Über die Konfiguration des Tasks können spezielle Analysepakete ausgewählt oder die Ausführung von Hilfsmodulen gesteuert werden.

Module

Einen Überblick über wichtigsten Bestandteile und die Lage der Module (gelb) sowie des Samples (blau) findet sich in Abbildung 9.2. Es muss berücksichtigt werden, dass es verschiedene Möglichkeiten gibt, eine Sandbox Infrastruktur aufzubauen.

9.4.2 *Ablauf einer Analyse*

Ablauf

Die Analyse eines Samples umfasst 6 Schritte, die im Folgenden näher betrachtet werden und einen Überblick über den Ablauf einer Analyse geben.

(1) NEUER TASK. Über verschiedene Schnittstellen (vgl. Abschnitt 9.4.5) kann ein neuer Analyseauftrag angelegt werden. Dabei ist es möglich, mittels Optionen die verschiedenen Analyse-, Verarbeitungs- und Hilfsmodule für den jeweiligen Analysevorgang zu konfigurieren.

(2) VORBEREITUNG DER ANALYSE. Die virtuelle Maschine wird mithilfe eines Rollback zu einem Snapshot auf die Ausführung vorbereitet. Ebenfalls werden alle konfigurierten Hilfsmodule wie z. B. die Aufzeichnung von Netzwerkverbindungen auf dem Hostsystem gestartet.

(3) ANALYSEVORGANG. Die Ausführung gestaltet sich je nach ausgewähltem Analysemodul unterschiedlich. Bei einer ausführbaren Datei wird diese lediglich gestartet. Handelt es sich um eine Webadresse, muss ein Browser mit der gewählten Webadresse als Parameter geöffnet werden.

(4) BEENDIGUNG DER ANALYSE. Das Analyseende ist erreicht, wenn alle Prozesse beendet sind, die im Laufe der Ausführung gestartet wurden. Alternativ wird nach Ablauf eines Timers die Beendigung erzwungen. Nach Übertragung der Daten wird die virtuelle Maschine beendet.

(5) VERARBEITUNG DER MESSDATEN. Die Verarbeitungsmodule generieren Informationen aus den gemessenen Daten. In diesem Vorgang werden die Messdaten zu Analyseergebnissen aufgearbeitet. Die einzelnen Verarbeitungsmöglichkeiten werden in Abschnitt 9.4.3 näher betrachtet.

(6) BERICHTERSTATTUNG. Die Berichterstattung basiert auf den Ergebnissen der Verarbeitungsmodule und kann unterschiedlich ausgerichtet sein. Über die Schnittstellen der Sandbox (Abschnitt 9.4.5) können die Ergebnisse abgerufen und ggf. weiter verarbeitet werden.

Weiteres

Weitere Details zur Vorbereitung und zum Ablauf einer Analyse kann der Dokumentation [37] entnommen werden. Die Nutzung der Analyseergebnisse ist für das weitere Verständnis von wichtiger Bedeutung und wird aus diesem Grund genauer betrachtet.

9.4.3 *Verarbeitung der Messdaten*

Module

Im Folgenden wird ein Auszug von wichtigen Verarbeitungsmodulen gegeben, wie sie bereits in Abschnitt 9.4.1 erwähnt wurden. Die Module generieren aus den rohen Messdaten verwertbare Informationen, die anschließend auf verschiedene Weise in Berichten dargestellt bzw. kombiniert werden können. Die hier aufgeführten Listeneinträge entsprechen dem Namen des jeweiligen Verarbeitungsmoduls.

ANALYSISINFO. Dieses Modul sammelt Rahmeninformationen zur Analyse wie beispielsweise das Datum der Auftragserstellung, Messung und Berichterstattung sowie genutzte Module und gesetzte Optionen.

BEHAVIORANALYSIS. Wie in Abschnitt 9.2.4 beschrieben wurde, ist die Analyse des Verhaltens einer Schadsoftware von wichtiger Bedeutung für die dynamische Schadsoftwareuntersuchung. Diese Verhaltensinformationen werden im `Behavior`-Log abgebildet. In Quelltext 9.1 ist ein Beispiel[36] für ein solchen Logeintrag zu sehen. Dieses zeigt den Aufruf der bereits bekannten `CreateDirectoryW`-Funktion, welches anschließend als Profilverzeichnis für den Browser dient.

NETWORKANALYSIS. Neben dem Verhalten im Betriebssystem können auch Netzwerkaktivitäten wichtige Hinweise liefern. Die Ergebnisse des Netzwerkmitschnitts (pcap-Datei) werden analysiert und so aufbereitet, dass ein Überblick aller Kommunikationsbeziehungen möglich ist.

DROPPED. Während der Ausführung einer Software werden üblicherweise Dateien auf dem Dateisystem erstellt oder modifiziert. Nach Beendigung der Schreibvorgänge werden diese von der Sandboxumgebung gesichert und können zu einem späteren Zeitpunkt eingesehen werden. Die Verarbeitung stellt diese gebündelt zur Verfügung und werden über einen kryptografischen Hashcode indiziert.

MEMORY. Sofern konfiguriert, wird nach der Analyse ein Abbild des Arbeitsspeichers erhoben und mittels geeigneter Werkzeuge analysiert. Je nach Virtualisierungssystem kann dieser Speicher ohne Mitwirkung des Betriebssystems direkt vom Hypervisor ausgelesen werden.

9.4.4 *Ergebnisberichte*

Übersicht

Die aus Abschnitt 9.4.1 bekannten `Berichtsmodule` erzeugen aus dem Verarbeitungsergebnis, Abschnitt 9.4.3, verschiedene Berichtsformen. Die Sandbox lässt hierbei eine leichte Erweiterbarkeit zu. In diesem Abschnitt wird nur das Ergebnis im JSON-Format näher betrachtet, weil dieses ein nahezu vollständiges Analyseergebnis aller Verarbeitungsmodule in einer einheitlichen Datenstruktur bietet. Es enthält auf erster Ebene die folgenden Schlüsseleinträge:

- `info`,
- `signatures`,
- `target`,
- `buffer`,
- `dropped`,
- `behavior`,

36 Im JSON-Format – grundsätzlich können die Daten auch in anderer Weise abgebildet bzw. präsentiert werden.

- `debug`,
- `metadata`,
- `screenshots`,
- `network`,

Abschnitte

Die Einträge zu `info`, `dropped`, `behavior` und `network` wurden bereits in Abschnitt 9.4.3 erläutert. Bei `signatures` findet eine Einordnung der Messergebnisse in bestimmte Kategorien statt: z. B. API-Aufrufe, die für bestimmte Schadsoftwaretypen üblich sind. Sofern Datenfragmente sich nur kurzzeitig im Arbeitsspeicher oder auf der Festplatte befinden, sind diese in `buffer` vermerkt. Einträge in `debug` sind überwiegend Logausgaben der verwendeten Werkzeuge. Unter `metadata` werden alle Dateien gelistet, die während des Analysevorgangs von der Schadsoftware und von den Analysesystemen erstellt werden. Bildschirmaufzeichnungen sind dem Schlüsselwort `screenshots` untergeordnet.

```
{
    "category": "file",
    "status": 1,
    "stacktrace": [],
    "api": "CreateDirectoryW",
    "return_value": 1,
    "arguments": {
        "dirpath_r": "c:\\users\\lab\\appdata\\local\\temp\\
            tmpgrqnzo",
        "dirpath": "c:\\Users\\lab\\AppData\\Local\\Temp\\
            tmpgrqnzo"
    },
    "time": 1506351043.005202,
    "tid": 3124,
    "flags": {}
}
```

Quelltext 9.1: Beispielergebnis aus der Verarbeitung des `BehaviorAnalysis`-Moduls im JSON-Format.

9.4.5 *Schnittstellen*

Steuerung

Die Sandbox kann auf vielfache Weise angesteuert werden. Auch zur Betrachtung der Ergebnisse stehen verschiedene Mittel zur Verfügung. Diese teilen sich dabei in Kommandozeile, Webinterface und REST-API auf.

KOMMANDOZEILE. Über die `cuckoo`-Anwendung lassen sich Anweisungen an die Sandbox übertragen[37]. Die Anwendung dient vornehmlich zum Erstellen neuer Analyseaufträge; Ausgabe oder Verarbeitung von Analyseergebnissen sind nicht vorgesehen.

WEBINTERFACE. Die Weboberfläche dient der vollständigen Interaktion mit der Sandbox. Neben der Erstellung von neuen Analyseaufträgen ist die Verfolgung aktueller Vorgänge sowie die Ergebnisse vorheriger

37 http://docs.cuckoosandbox.org/en/2.0.3/usage/submit/, abgerufen am 17.02.2018.

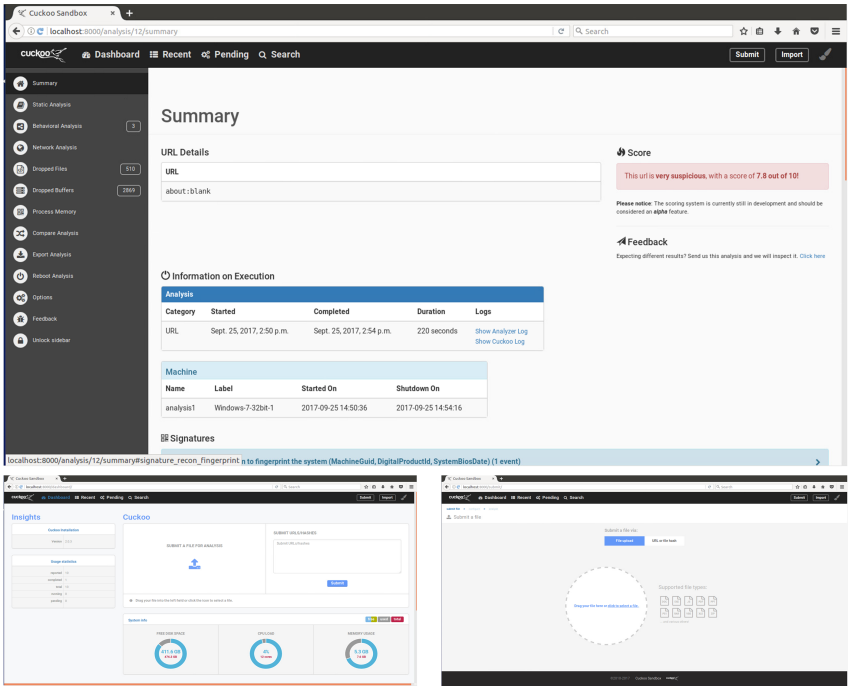

Abbildung 9.3: Webinterface der Cuckoo Sandbox. Oben: Zusammenfassung der Analyse / links: Dashboard / rechts: neuer Analyseauftrag.

Analysen einsehbar. Die Webanwendung lässt sich auch als uWSGI[38] Container in einen bestehenden nginx Webserver integrieren. Auszüge der Oberfläche zeigt Abbildung 9.3.

REST API. Bei REST (Representational State Transfer) handelt es sich ebenfalls um eine webbasierte Schnittstelle. Diese Schnittstelle dient der programmatischen Interaktion mit der Sandbox. Über dieses Interface ist die Erstellung bzw. Übertragung neuer Aufträge sowie das Auslesen der Ergebnisberichte im JSON-Format möglich.

9.5 Die Sandbox als Messwerkzeug zur Analyse von Web-Tracking

Verhalten

Bisher wurde die Sandbox als Analysewerkzeug von potenzieller Schadsoftware betrachtet. In der Literatur finden sich erste Ansätze, die das Verhalten des Browsers berücksichtigen. Beispielsweise verwenden Acar et al. [2] das Tool *strace* zur Detektion von Flashcookies durch Überwachung der schreibenden Zugriffe auf ein Verzeichnis. Bisher wurde kein Versuch unternommen, eine solche Analyseumgebung, z. B. mit Rückgriff auf eine Sandbox, zu standardisieren.

Voruntersuchung

In einer publizierten Voruntersuchung [189] wurde der Internet Explorer 8 in einer Sandbox-Umgebung während des Abrufs von populären Webseiten überwacht. Ziel war die Erfassung aller Speicherformen, insbesondere

38 https://uwsgi-docs.readthedocs.io/en/latest/, abgerufen am 17.02.2018.

der zum damaligen Zeitpunkt üblichen Flash-Cookies der (Adobe) Flash-Erweiterung. In der Untersuchung zeigte sich eine hohe Invasivität des Browsers, welche sich durch Speicherungen in Systemverzeichnisse und Veränderungen der Windows-Registry bemerkbar machten.

Ergebnis

Der Einsatz einer Sandbox zur Browserüberwachung erwies sich demnach als effektives Werkzeug um:

- eine vollständige Auflistung der Lese- und Schreibaktivitäten des Browsers zu generieren und somit einen Überblick der Invasivität zu erhalten und
- durch einen Snapshot der virtuellen Maschine das System in den ursprünglichen Zustand zurückzuführen.

Werkzeuge

Im Folgenden werden zunächst bestehende Werkzeuge zur Messung von Web-Tracking vorgestellt. Es ist zu erkennen, dass diese den Browser modifizieren und erweitern, jedoch keine verhaltensbasierten Analysen durchführen. Erste Überlegungen zum Mehrwert einer solchen Analysemethode stellen den Übergang zu Kapitel 10 dar.

9.5.1 *Bestehende Web-Tracking Messwerkzeuge*

Übersicht

Zunächst findet eine genauere Betrachtung bestehender Messwerkzeuge statt. In Englehardt et al. [48] werden verschiedene „Web privacy measurement platforms“ vorgestellt: FPDetective, Adfisher, WebXray, FourthParty und OpenWPM.

FPDetective

Wie von Acar et al. [1] publiziert wurde, ist der FPDetective[39] speziell zur Suche und Analyse von Fingerprint-Verfahren, wie sie in Abschnitt 7.4 beschrieben wurden, entwickelt worden. Dafür kommen verschiedene Browsertypen (PhantomJS und Chromium) und ein Proxy zur Aufzeichnung der Netzwerkaktivitäten zum Einsatz.

Adfisher

Der Adfisher von Datta et al. [40, 39] wurde speziell zur Untersuchung von Wechselwirkung zwischen Nutzeraktivitäten und Onlinewerbung entwickelt. Wie beim FPDetective handelt es sich also nicht um ein allgemeines Messwerkzeug, sondern es wurde zur Beantwortung einer konkreten Forschungsfrage entwickelt.

WebXRay

Bei WebXRay[40] handelt es sich um ein von Timothy Libert entwickeltes Werkzeug zur Ermittlung von Drittparteieinbettungen auf Webseiten. Ver-

39 https://www.cosic.esat.kuleuven.be/fpdetective/, abgerufen am 22.02.2018.
40 https://github.com/timlib/webXray, abgerufen am 22.02.2018.

wendet wurde der Browser PhantomJS, welcher durch eine in der Programmiersprache Python entwickelte Anwendung gesteuert wird. Eine Liste von Adressen (URIs) werden von dieser sequenziell abgearbeitet – die anfallenden Netzwerkdaten werden in eine MySQL-Datenbank geschrieben und anschließend ausgewertet.

Es ist zu berücksichtigen, dass PhantomJS nicht alle Funktionalitäten eines etablierten User-Agents (bzw. dessen Renderers) unterstützt. Die Entwickler führen selbst aus[41], dass insbesondere im Bereich der Visualisierung (z.B. bei 3D-Inhalten) zahlreiche Funktionen nicht implementiert wurden.

Fourthparty

Bei Fourthparty[42] handelt es sich um eine von der Community entwickelte Open-Source Software, die von Jonathan Mayer [115] initiiert wurde. Unter Verwendung der Browserautomatisierung Selenium wird der Mozilla Firefox Browser unter Ergänzung von Addons zur Erhebung von Daten verwendet. Da es sich um eines seit 2015 nicht länger weiterentwickeltes Projekt handelt, und funktional vollständig von seinem Nachfolger OpenWPM ersetzt wurde, wird dieses nicht weiter betrachtet.

OpenWPM

OpenWPM[43] ist ein Framework zum Vermessen von Webseiten – insbesondere für Web-Tracking. Es kommt in diversen Publikationen zum Einsatz [2, 49, 94, 111]. Vorgestellt und evaluiert wird das Werkzeug von Englehardt und Narayanan [48].

Die Automatisierung des Mozilla Firefox Browsers erfolgt mit Selenium und die Datenerhebung wird durch eine eigene Firefox-Erweiterung `openwpm.xpi`[44] umgesetzt. Dies ermöglicht die feingranulare Messung folgender Daten:

- Header von HTTP-Anfragen und Antworten,
- JavaScript-Methodenaufrufe,
- Inhaltsdaten von HTTP-Antworten,
- Cookies, Flashcookies[45] und
- Screenshots.

Die Speicherung erfolgt in einer Datenbank, wodurch die Auswertung der Daten erleichtert wird. OpenWPM lässt sowohl stateless als auch stateful

41 http://phantomjs.org/supported-web-standards.html, abgerufen am 02.04.2018.
42 http://fourthparty.info/, abgerufen am 22.02.2018.
43 https://github.com/citp/OpenWPM/tree/63610ca485d160f159babbe6a26634cf1a896083, abgerufen am 22.08.2018.
44 https://github.com/citp/OpenWPM/tree/63610ca485d160f159babbe6a26634cf1a896083/automation/Extension/firefox, abgerufen am 22.02.2018.
45 Wird mittels Überwachung des dafür bekannten Ordners im Dateisystem umgesetzt.

Crawls zu: Gemeint ist damit, dass das Browserprofil während den Webseitenbesuchen erhalten bleibt (stateful) oder jedes Mal verworfen wird (stateless). Im letzten Fall wird sich auf die Löschung des Browserprofils und einiger fest definierter Verzeichnisse beschränkt.

Einschränkungen des Werkzeugs zeigen sich an den Stellen, an denen die grundsätzliche Erweiterbarkeit des Browsers an ihre Grenzen stößt. So besteht keine Möglichkeit, die Inhaltsdaten von DNS- oder OCSP-Abfragen abzufangen oder einzusehen. Auch HTTP-Header von Bildern, die sich bereits im Cache befinden, werden nicht protokolliert[46].

9.5.2 *Vorteile der Sandbox*

Abgrenzung

Die vorgestellten Werkzeuge zielen darauf ab, umfangreiches Datenmaterial durch einen Webseitenbesuch zu sammeln. Auf diese Weise soll die Webseite möglichst exakt vermessen und anschließend bezüglich bekannter Web-Tracking-Verfahren ausgewertet werden. Die Werkzeuge verwenden Schnittstellen des Browsers für die durchgeführten Messungen. Eine genauere Betrachtung der Verhaltensweise des Browsers findet nicht statt.

Speicherungen

Wie bereits in Abschnitt 9.3 beschrieben wurde, ermöglicht eine Sandbox das Erfassen von Lese- und Schreibvorgängen einer Applikation auf das umliegende System. Mithilfe einer Sandbox, wie sie in Abschnitt 9.4 ausführlich beschrieben wurde, ist es möglich, alle Aufrufe der Windows API während dem Abruf einer Webseite zu verfolgen. Speicherungen zeigen sich durch Aufruf der dafür vorgesehen Systemfunktionen. Bei speicherbasierten Web-Tracking-Verfahren, bekannt aus Abschnitt 7.3, werden identifizierende Merkmale auf dem System des Besuchers abgelegt. Diese Speichervorgänge können durch eine Prozessüberwachung registriert werden.

Verhalten

Ebenfalls ist eine Überwachung des Netzwerkverkehrs möglich. Dies umfasst nicht nur die Aufrufe durch den Browser, sondern alle, die in irgendeiner Weise das System verlassen. So wird der Analysefokus vom Browser zum Gesamtsystem verlagert. Auch Abfragen oder Änderungen der Windows-Registry können möglicherweise weitere Anhaltspunkte bzgl. Web-Tracking liefern. Die zusätzlichen Möglichkeiten der Virtualisierung können wertschöpfend eingesetzt werden. Durch Rücksprung zu einem Snapshot ist eine vollständige Bereinigung des Systems möglich. Auf diese Weise werden unberücksichtigte Effekte ausgeschlossen, so dass eine annähernd[47] vollständige Reproduzierbarkeit der Ausgangssituation möglich ist.

Schnittstellen

Die Schnittstellen und der modulare Aufbau der Sandbox ermöglichen eine leichte Erweiterbarkeit. Module können zur Verarbeitung angepasst und die Berichterstattung kann entsprechend erweitert werden. Aus diesem

46 „Note: request and response headers for cached content are also saved, with the exception of images. See: Bug 634073.“, https://bugzilla.mozilla.org/show_bug.cgi?id=634073, abgerufen am 22.02.2018.

47 Ausnahmen werden von äußeren Faktoren wie beispielsweise der Uhrzeit begründet.

Grund kommt die Sandbox als Werkzeug in Betracht, die den Browser bei Abruf einer Webseite überwacht, alle Interaktionen mit dem umliegenden System protokolliert und anschließend auf mögliche Tracking-fähige Speicherungen zu untersuchen.

10 ENTWURF UND IMPLEMENTIERUNG VON DISTRACK

Zusammenfassung: Mit der Entwicklung des Analyse-Frameworks DisTrack sollen die Vorteile einer Sandbox-basierten Analyse untersucht werden. An das Design werden konkrete Anforderungen gestellt, die auch in der darauffolgenden Implementierung berücksichtigt werden. Dafür notwendige Anpassungen der bestehenden Sandbox werden erläutert und dokumentiert. Ziel dieses Kapitels ist, den Entwicklungsprozess von DisTrack aufzuzeigen und die getroffenen Designentscheidungen zu begründen.

10.1 EINLEITUNG

Motivation

Die in Abschnitt 9.5 beschriebene Voruntersuchung [189] hat gezeigt, dass der Browser zu umfangreichen Systemänderungen in der Lage ist. Abbildung A.1 in Anhang A gibt einen Überblick über die Änderungen an der Windows-Registry, die durch den Internet Explorer 8 bedingt sind. Diese Erkenntnis soll motivieren, die Auswirkungen des Browsers in Bezug auf Speicherungen, die für speicherbasiertes Tracking genutzt werden können, näher zu untersuchen.

Anforderungen

Allein die Öffnung von Webseiten sowie die Analyse des Browsers sind jedoch kein ausreichendes Werkzeug für eine strukturierte Analyse. Benötigt wird ein Framework, das

- die Modellierung von neuen Tests ermöglicht,
- umfangreiche Messmöglichkeiten bietet,
- die Sandbox entsprechend dieser Testdurchläufe steuert und
- eine Auswertung der Daten ermöglicht.

DisTrack

Zu diesem Zweck wird das Analyseframework DisTrack[1] entwickelt. Dieses ermöglicht eine strukturierte Erfassung und Auswertung der Prozessaktivitäten. Grundsätzlich ist eine Kombination der Sandbox-Umgebung mit bestehenden Analysewerkzeugen aus Abschnitt 9.5.1 möglich. Aus diesem Grund handelt es sich bei DisTrack nicht um ein weiteres Analysewerkzeug für Web-Tracking, sondern um ein Framework zum Einsatz einer Sandbox in Web-Tracking-Analysen. Damit ist eine Erweiterung zu bestehenden Analysewerkzeugen möglich, ohne mit diesen in Konkurrenz zu stehen.

Aufbau

Das Kapitel ist so organisiert, dass zu Beginn die Methodik in Abschnitt 10.2 näher betrachtet wird und in Abschnitt 10.3 Anforderungen aufgestellt werden. Anschließend findet ein Entwurf des Frameworks statt (Abschnitt 10.4), das darauffolgend umgesetzt wird (Abschnitt 10.5). Dabei müssen insbesondere die spezifizierten Anforderungen berücksichtigt werden. Aufgrund

1 DisTrack ist eine Kombination der Wörter Dissection (=Sezierung) und Tracking.

des besonderen Analysefokus, werden ebenfalls Modifikationen an der bestehenden Sandbox-Implementierung notwendig sein, die in Abschnitt 10.6 dokumentiert werden. Die Verknüpfung von DisTrack mit OpenWPM wird in Abschnitt 10.7 beschrieben. Genauere Informationen zu Versionsständen von genutzten Anwendungen und Bibliotheken sind in Abschnitt 10.8 einsehbar. Die Evaluation des Werkzeugs und die Prüfung der Anforderungen ist für Kapitel 11 vorgesehen.

10.2 Methodik

10.2.1 *Forschungsfragen*

Forschungsfrage

Nach der Vorstellung der Sandbox in Kapitel 9 stellt sich die Frage, ob diese im Bereich Web-Tracking sinnvoll eingesetzt werden kann. Es ist offen, mit welcher Technik dies umgesetzt werden kann. Auch ist ungeklärt, welche Vor- und Nachteile sich daraus ergeben. Auf dieser Basis lässt sich die Forschungsfrage RQ-4 stellen:

> **Forschungsfrage RQ-4** Welchen Mehrwert bietet die verhaltensbasierte Analyse des Browsers zur Erkennung von Web-Tracking-Methoden?

10.2.2 *Forschungsmethodik*

DSR-Zyklus

Entwurf, Entwicklung und Evaluation des Frameworks (Artefakt) werden auf Basis des Design Science Research (Vaishnavi und Kuechler [187]) durchgeführt. Die Methodik wurde bereits in Abschnitt 1.3 vorgestellt. Das Modell umfasst die fünf Phasen: Awareness of Problem, Suggestion, Development, Evaluation und Conclusion. Die Phasen sind in Abbildung 10.1 dargestellt und werden im Folgenden genauer betrachtet.

Awareness of problem In Kapitel 9 wurde die Funktionsweise einer Sandbox beschrieben. Bei einer genaueren Betrachtung zeigen sich Ähnlichkeiten in der Arbeitsweise von Schadsoftware und Tracking-Verfahren. In Abschnitt 9.5.2 wurden deshalb erwartete Vorteile durch den Einsatz einer Sandbox angedeutet.

Suggestion In der Konzeptionsphase werden Anforderungen formuliert, welche die gesuchte Implementierung erfüllen muss. Dies wird in Abschnitt 10.3 vorgenommen. Anhand dieser Anforderungen und den gegebenen Rahmenbedingungen soll ein Framework zur Durchführung solcher Analysen entworfen werden (Abschnitt 10.4).

Development Die Entwicklung des Frameworks richtet sich nach dem Design der Konzeptionsphase und wird in Abschnitt 10.5 beschrieben. Um den aktuellen technischen Stand zur verhaltensbasierten Analyse von Software zu berücksichtigen, wird eine bestehende Implementierung verwendet, die sich in der Praxis bewährt hat.

EVALUATION Die Evaluierung unterteilt sich in drei Teile. Zunächst werden Werkzeug- und Vergleichstests durchgeführt, welche die korrekte Arbeitsweise des Werkzeugs belegen. Anschließend werden durch Definition, Umsetzung und Ausführung von Analysemodellen die Alleinstellungsmerkmale von DisTrack hervorgehoben. Abschließend findet eine Prüfung der Anforderungen aus der Konzeptionsphase statt. Die Evaluationsmethodik wird in Abschnitt 11.1 eingehender beschrieben.

CONCLUSION Der Abschluss wird durch die Schlussfolgerungen in Abschnitt 11.8 gebildet. Das Fazit ist in Kapitel 12 zu finden. Dieses umfasst ebenfalls die anderen Teile dieser Dissertation.

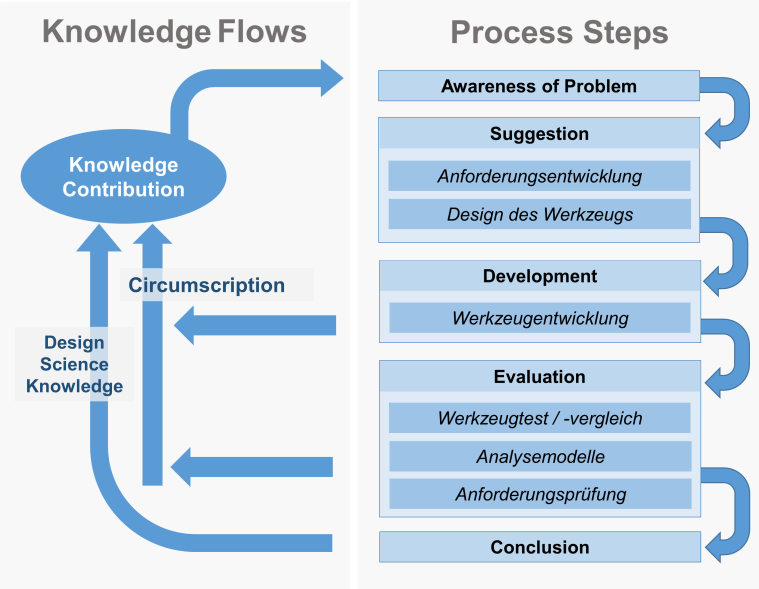

Abbildung 10.1: Konkretisierung des DSR-Zyklus, vorgestellt von Vaishnavi und Kuechler [187]. Eigene Ergänzungen sind deutsch und kursiv ausgezeichnet.

Zusammenfassung

Zusammenfassend zielt Kapitel 10 auf die Werkzeugentwicklung ab und klärt die Frage nach der Technik. Kapitel 11 dient der Evaluierung des Werkzeugs und soll die Vor- und Nachteile dieser speziellen Analysetechnik hervorbringen.

10.3 ANFORDERUNGEN

Bestehende Anforderungen

Von Englehardt et al. [48] wurden vier Anforderungen an das Messwerkzeug OpenWPM definiert und auf deren Basis evaluiert: Stabilität (Stability), Vollständigkeit (Completeness), Ressourceneinsatz (Reource Usage) und Allgemeinheit (Generality). Dabei ist zu berücksichtigen, dass das Werkzeug für quantitative Zwecke entworfen wurde.

Anforderungen an DisTrack

Obwohl sich die Zielsetzungen des DisTracks-Framework von den verwandten Arbeiten unterscheiden, werden die Anforderungen von Engle-

hardt et al. berücksichtigt und um die Forderung nach „Authentizität“ erweitert:

(B-1) **Stabilität**. Insbesondere wenn das Werkzeug für quantitative Studien eingesetzt wird, die üblicherweise eine hohe Anzahl von Webseiten umfassen, muss die Stabilität gewährleistet sein. Ebenfalls muss dafür gesorgt sein, dass fehlerhafte Auswertungen nachträglich erkannt und nicht berücksichtigt werden, so dass ggf. eine Wiederholung der Messung möglich ist.

(B-2) **Vollständigkeit**. Zum Vermessen von Webseiten mit den in Abschnitt 9.5.1 beschriebenen Werkzeugen werden häufig spezielle User-Agents eingesetzt (z. B. PhantomJS in WebXRay). Dabei bleiben möglicherweise Effekte unberücksichtigt, die nur bei „echten“ Browsern eintreten[2]. Eine (vollständige) Messung von Web-Tracking muss mit den gleichen Anwendungen (Browsern) unternommen werden, die auch von Nutzern im privaten und beruflichen Alltag eingesetzt werden.

(B-3) **Effizienz**. Im Zuge der retrospektiven Analyse wurde in Abschnitt 4.5 die Effizienz als wichtige Anforderung für quantitative Studien identifiziert. Wichtig ist, dass keine vermeidbaren Verzögerungen entstehen oder ggf. Wege gesucht werden, diese zu kompensieren.

(B-4) **Allgemeinheit**. Ein Messwerkzeug sollte allgemein für verschiedene Formen von Analysen nutzbar und nicht auf ein spezielles Nutzungsszenario zugeschnitten sein. Auf Messdaten müssen mit einfachen Mitteln zugegriffen und programmatisch verarbeitet werden können. Je nach Art der Analyse muss das Werkzeug angepasst und auf einen bestimmten Fokus ausgerichtet werden können. Solche Veränderungen müssen mit geringem Aufwand zu erbringen sein.

(B-5) **Authentizität**. In bestehenden Messwerkzeugen wird der Browser verändert bzw. durch Addons erweitert, um Webseitenaktivitäten zu messen (vgl. Fourthparty oder OpenWPM). Dies bedeutet stets eine Veränderung des Browsers. Bei Nutzung von Prozessüberwachungsmethoden das Vermeiden möglicher Verzerrungen durch Erweiterungen wichtig. Die zur Messung bedingten Veränderungen am Browser sollen keine abweichenden Resultate in den Messergebnissen bewirken.

10.4 Entwurf von DisTrack

Grundlegende Ziele

Um den Überlegungen zum Entwurf von DisTrack besser folgen zu können, wird zunächst der geplante Einsatzzweck erläutert. Das Framework soll ermöglichen, Sandbox-basierte Tests zu erstellen und die gemessenen Daten

2 Englehardt et al. [48] zeigen anhand des Beispiels Dropbox, wie Webseiten sich je nach Browsertyp unterschiedlich verhalten.

zu vergleichen bzw. flexibel zu analysieren. Ein Test besteht beispielsweise daraus eine bestimmte Webseite zu öffnen, anschließend einen Rücksprung der virtuellen Maschine (Snapshot) durchzuführen und die Webseite in einem zweiten Analyseschritt noch mal zu öffnen. Im Anschluss können die gemessenen Daten auf Gemeinsamkeiten und Unterschiede untersucht werden. In Bezug auf Web-Tracking sind alle Unterschiede von besonderem Interesse. Es ist wichtig einzusehen, dass der Untersuchungsgegenstand nicht nur die Webseite ist, sondern auch der Browser während dem Öffnen der Webseite. Dies unterscheidet DisTrack von bisherigen Analysemethoden.

Weitere Ziele

Das hier geschilderte Nutzungsszenario ist ein sehr einfaches. Doch grundsätzlich sollen mit DisTrack auch komplexere Testmöglichkeiten möglich sein, in denen der Systemzustand (Snapshots) und das Browserprofil kombiniert werden. Die Vorstellung von komplexeren Nutzungsszenarios ist für Kapitel 11 vorgesehen. Neben der bereits beschriebenen Testmöglichkeit wurden in der Einleitung 10.1 des Kapitels drei weitere Ziele des DisTrack-Frameworks festgelegt. Zunächst muss eine flexible Messung von Browseraktivitäten bei Abruf von Webseiten gegeben sein. Um die beschriebenen Tests umzusetzen, ist außerdem die Steuerung des Ablaufs erforderlich. Diese Ablaufsteuerung muss ebenfalls vom Framework übernommen werden, soweit dies nicht von der Sandbox erfüllt wird. Abschließend werden die gewonnenen Daten ausgewertet.

10.4.1 *Datenmessung*

Datenarten

Daten, die zur Analyse von Web-Tracking aus bestehenden Werkzeugen bezogen werden, unterscheiden sich grundsätzlich von denen, die durch eine Sandbox erzeugt werden. Innerhalb der Anwendung stehen die Interaktionen des Browsers mit der Webseite im Vordergrund: abgerufen Ressourcen, Laufzeiten, Ausführungen von JavaScript und Gestaltungsregeln. Bei einer Messung des Browsers innerhalb der Sandbox sieht man wesentliche Interaktionen mit dem umliegenden System: geöffnete Dateien, geladene Bibliotheken, ausgelesene Registrierschlüssel, Benutzereingaben und Kommandozeilenargumente. Ausgaben sind geschriebene, modifizierte, gelöschte Dateien und Netzwerkaktivitäten.

Erweiterungen

Auf welche Weise eine Sandbox an diese Daten gelangt, wurde bereits in Kapitel 9 beschrieben. Auf der Browserseite ist hierfür die WebExtensions API einschlägig[3]: sie wird aktuell von marktüblichen Browsern (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera) unterstützt. Mit solchen Erweiterungen ist das Abfangen und Aufzeichnen von HTTP-Anfragen[4] oder das Auslesen von Cookie-Informationen[5] möglich. Anders als bei proprietä-

3 Die Messwerkzeuge setzen auf proprietäre Erweiterungen, da es sich bei WebExtensions noch um einen recht jungen Standard handelt.

4 https://developer.mozilla.org/en-US/Add-ons/WebExtensions/Intercept_HTTP_requests, abgerufen am 01.03.2018.

5 https://developer.mozilla.org/en-US/Add-ons/WebExtensions/API/cookies, abgerufen am 01.03.2018.

ren Erweiterungen existiert für WebExtensions ein Berechtigungskonzept[6], um deren Zugriffsrechte zu regeln.

Nachteile

In den vorgestellten Werkzeugen aus Abschnitt 9.5 werden die Messungen durch solche Browsererweiterungen unterstützt. Dies ist leicht nachvollziehbar, weil auf diese Weise die internen Vorgänge mit geringem Aufwand erfasst werden können. Bei einer Erweiterung des Browsers kann bei einer Sandbox-basierten Analyse nicht zugeordnet werden, welche Systemaktivitäten durch den Browser und welche durch die Browsererweiterungen hervorgerufen werden. So kann die Erweiterung des Browsers zum Auslesen von Messdaten zu einer möglichen Verhaltensänderung führen. Die Erfassung von Messdaten unter Einsatz von Browsererweiterungen ist eine von der üblichen Nutzung abweichende Verwendung. Aus diesem Grund muss die Authentizitäts- (B-5) und die Allgemeinheitsanforderung (B-4) als konkurrierend betrachtet werden.

Anforderung

Für eine verhaltensbasierte Analyse sollte der Browser möglichst unmodifiziert bleiben. Aus diesem Grund ist es vorteilhaft, wenn das genutzte Messwerkzeug ohne Modifikation oder Erweiterung des Browsers auskommt.

10.4.2 *Modellierung von Tests*

Profil

Der Kontext des Browsers wird maßgeblich durch das Browserprofil bestimmt. Englehardt und Narayanan [48] definieren den Begriff „Stateless Crawl“ für eine Untersuchungsart, in der für jede Webseitenüberprüfung ein neues temporäres Browserprofil angelegt wird. Bei einem „Stateful Crawl“ wird ein bestehendes Browserprofil wiederverwendet und durch weitere Webseitenaufrufe ergänzt.

System als Kontext

Durch die Nutzung einer Sandbox und die Möglichkeiten der Virtualisierung, kann der Zustand des umliegenden Systems kontextbildend berücksichtigt werden. Damit geht die Fähigkeit einher, sowohl das Browserprofil als auch den Systemzustand zu sichern und wiederzuverwenden. Aus diesem Grund wird mit Abbildung 10.2 eine Notation eingeführt, die eine Trennung zwischen Profil- und Systemzustand visualisiert. Dabei können diese Zustandsarten auch getrennt voneinander in neuen Tests kombiniert werden.

Testmodell

Das Beispiel zeigt (obere Hälfte in Abbildung 10.2), wie in Schritt 1 ein neuer, bisher ungenutzter System- und Profilzustand erzeugt wird. Innerhalb der Schritte findet eine Ausführung des Browsers, das Öffnen einer oder mehrerer Webseiten oder sonstige Browserinteraktionen statt. Sie spiegeln Analysedurchläufe der Sandbox wider. Nach Ausführung von Schritt 1 muss das Browserprofil gesichert werden, um diesen in Schritt 2 verwenden zu können. Der Systemzustand kann verworfen werden, da er in keinem anderen Schritt referenziert wird. Nach Ausführung von Schritt 2 müssen sowohl der System- als auch Profilzustand gesichert werden, da diese für Schritt 3

6 https://developer.mozilla.org/en-US/Add-ons/WebExtensions/manifest.json/permissions, abgerufen am 01.03.2018.

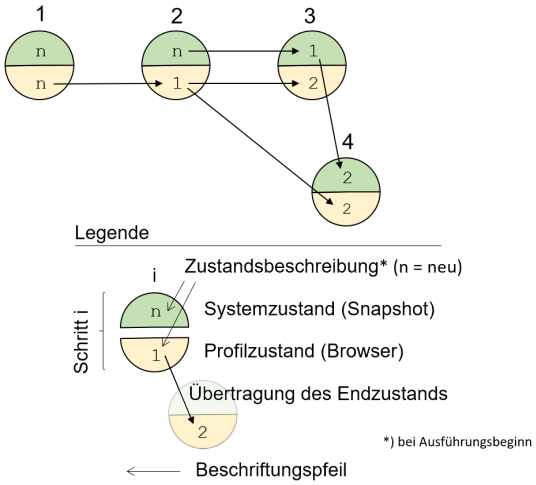

Abbildung 10.2: Notation und Beispiel von Tests.

notwendig sind. Nach Abschluss der Untersuchungen von Schritt 3 muss der Systemzustand gesichert werden, da in Schritt 4 das Ergebnisprofil aus Schritt 2 verwendet wird, wie an der Transition L1 zu erkennen.

Wie Tests erstellt und von DisTrack verarbeitet werden, wird im Implementierungsabschnitt 10.5.3 näher betrachtet.

10.4.3 *Steuerung der Sandbox*

Schnittstellen

Über die Schnittstellen der Sandbox müssen Steuerungsmöglichkeiten zur Verfügung stehen, die DisTrack zur Durchführung der Tests benötigt. Alternativ muss die Sandbox um diese Möglichkeiten erweitert werden. Dazu zählen folgende Basisoperationen:

- die Erstellung von neuen Analysen (Submit),
- die Übermittlung von Konfigurationspräferenzen wie beispielsweise Timeout-Einstellungen,
- die Aktivierung und Deaktivierung von Hilfsmodulen,
- die Durchführung der Messung und
- das Bereitstellen von Messergebnissen über die Schnittstellen der Sandbox.

Weitere Funktionen

Neben diesen Kernaufgaben muss die Sandbox weitere Funktionen zur Verfügung stellen, um die in Abschnitt 10.4.2 beschriebenen Testmöglichkeiten zu ermöglichen. Weitere Funktionen sind:

- die Auswahl, Erstellung und Löschen von Snapshots der virtuellen Umgebung,
- das Bereitstellen von Applikationsdaten, bspw. einem gegebenen Browserprofil,
- die Sicherung zusätzlicher Applikations- und Systemdaten, bspw. des neuen Browserprofils oder des Zustandes von Systemdateien.

10.4.4 *Auswertungsumgebung*

Aufbau

Sowohl durch die Messungen der Sandbox als auch durch den Browser werden umfangreiche Daten erzeugt. Diese müssen in Abhängigkeit vom jeweils definierten Test gemäß Abschnitt 10.4.2 verarbeitet werden. Diese Daten entstehen in jedem Analyseschritt eines Tests, wie er beispielhaft in Abbildung 10.2 dargestellt ist. Die Auswertung muss sich an diesen Analyseschritten orientieren, weshalb eine Verschränkung des Analyse- und Auswertungsmodells gegeben. Daher empfiehlt sich, dass die Analyse in der gleichen Umgebung wie die Testmodellierung durchgeführt wird.

Anforderung

Das System darf die Möglichkeiten der Auswertung nicht einschränken, da es sich gemäß der Allgemeinheitsanforderung (B-4) um ein allgemeines Werkzeug handeln soll. Infolgedessen die Auswertung mit einer Programmiersprache (Turing-Vollständig) durchführbar sein. Das Framework kann durch bereitgestellte Programmbibliotheken die Auswertung der Ergebnisse unterstützen, indem es Methoden zur Generierung liefert.

10.5 IMPLEMENTIERUNG VON DISTRACK

Überblick

Während das Design von DisTrack von der verwendeten Sandboxumgebung losgelöst ist, muss sich eine Implementierung nach der gegebenen Sandbox richten. In Abschnitt 9.3.1 wurde bereits erwähnt, dass nach aktuellem Stand nur die Cuckoo Sandbox hinreichend flexibel und anpassbar ist. Aus diesem Grund fällt die Wahl auf die Cuckoo Sandbox, an der sich die DisTrack-Applikation anknüpft. So wie die Cuckoo Sandbox wird DisTrack in der Programmiersprache Python implementiert.

Modifikationen an Cuckoo

Einige der hier beschriebenen Funktionen sind nur durch Anpassungen der Sandbox möglich und werden zur Implementierung von DisTrack vorausgesetzt. Grundsätzlich sind diese Modifikationen auch Teil des Implementierungsprozesses. Um die Übersicht zu wahren und DisTrack strikter von Cuckoo trennen zu können, wurden diese in den eigenen Abschnitt 10.6 ausgelagert.

Aufbau

In Abbildung 10.3 ist der Aufbau von DisTrack, der Cuckoo Sandbox und der virtuellen Maschine exemplarisch dargestellt. DisTrack besteht aus vier Komponenten: Methoden und Berichte, Datenspeicherung, Ablaufsteuerung und WebChecker.

Komponenten

Der Aufbau der Cuckoo Sandbox und die dazugehörigen Komponenten wurden bereits in Abschnitt 9.4 beschrieben. Der WebChecker ist eine optionale Komponente und kann durch andere Messwerkzeuge ausgetauscht werden. OpenWPM ist ohne Einschränkung in einer Sandbox ausführbar, wie in Abschnitt 10.7 beschrieben wird. Auf diese Weise lassen sich die Vorteile der Messapplikation mit den zusätzlichen Messungen der Sandbox kombinieren. Der im Folgenden beschriebene WebChecker ermöglicht die Durchführung von Messungen, ohne eine Änderung oder Erweiterung am Browser vorzunehmen.

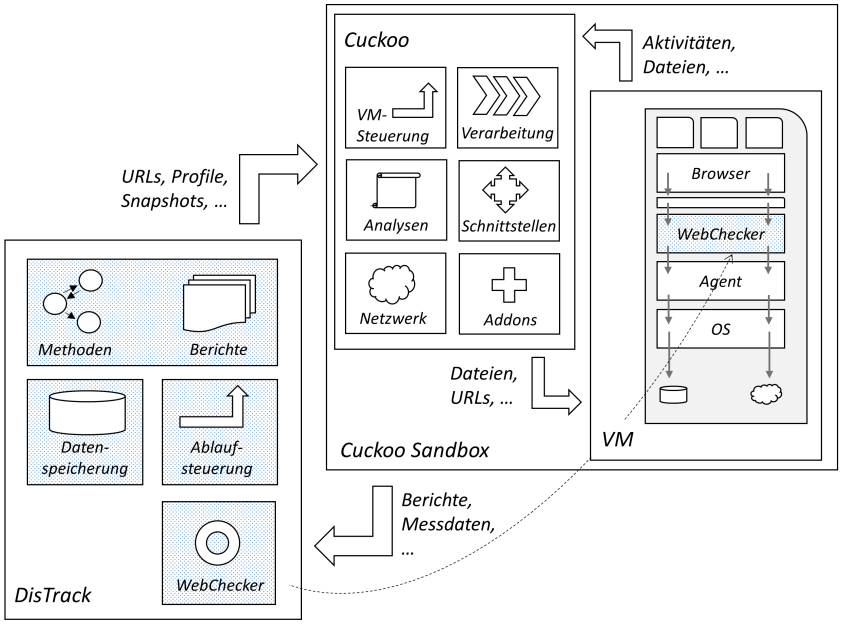

Abbildung 10.3: Aufbau des Framworks.

Übersicht

Die blau schraffierten Komponenten in Abbildung 10.3 werden in den folgenden Abschnitten 10.5.1-10.5.4 näher beschrieben. Sie orientieren sich stets an den Erkenntnissen der Entwurfsphase und den Anforderungen aus Abschnitt 10.3.

10.5.1 *WebChecker*

Messweise

Der WebChecker ist das Bindeglied zwischen DisTrack und Browser und ermöglicht eine nichtinvasive Messung von Webseiten. So werden nur Funktionen des Browsers zur Durchführung der Messung verwendet, ohne diesen zu verändern oder erweitern.

Wahl des Browsers

Vor- und Nachteile

Für die hier betrachtete Analyseform kann jeder marktübliche Browser verwendet werden. In der bereits veröffentlichten Voruntersuchung [189] wurde der Internet Explorer 8 auf einem Windows 7 Betriebssystem eingesetzt. Bei dieser Umsetzung war keine weitere Steuerung des Browsers möglich. Soll eine solche Steuerung umgesetzt und darüber hinaus zusätzliche Informationen vom Browser erhoben werden, müssen die integrierten Funktionen des Browsers betrachtet werden. Eine Übersicht[7] aktueller Browser (Microsoft Edge, Mozilla Firefox, Google Chrome) zeigt, dass alle einen Export von HTTP-Daten ermöglichen, allerdings nur im Mozilla Firefox automati-

7 https://toolbox.googleapps.com/apps/har_analyzer/, abgerufen am 26.03.2018.

siert werden kann. Aus diesem Grund fällt die Wahl auf den Mozilla Firefox Browser, der im Folgenden als Messwerkzeug genutzt wird.

Automatisierung

Steuerungsmethode

Zur Durchführung der in Abschnitt 10.4.2 beschriebenen Tests muss eine Automatisierung des Browsers ermöglicht werden. Im Zuge der retrospektiven Analyse wurde in Abschnitt 5.2.1 eine Übersicht verschiedener Automatisierungstechnologien gegeben. Der Einsatz von Selenium[8], wie es auch in OpenWPM verwendet wird, erwies sich als üblich, sofern ein normaler Browser zum Einsatz kommen soll. Selenium ermöglicht die programmierbare Steuerung des Browsers.

Selenium

In früheren Versionen ($\leq$ 3.0) wäre der Einsatz von Selenium problematisch gewesen, da eine Automatisierung mit einer Browsererweiterung notwendig gewesen wäre. In dieser Version wurde der Mozilla Firefox mit einem vorinstallierten WebDriver-Plugin gestartet, welches die Steuerung von außen ermöglicht. Dies stellt jedoch einen Eingriff in den Browser dar, der vermieden werden sollte. Seit Mozilla Firefox 47.0.1 kann der Browser über das Marionette-Protokoll[9] ferngesteuert werden, so dass Aktionen ohne die vorherige Installation einer Erweiterung ausführbar sind. Die Funktionen orientieren sich am WebDriver-Standard[10], der vom W3C für diesen Zweck entwickelt wurde und von Selenium seit Version 3.0 unterstützt wird. Das Bindeglied zwischen Selenium 3.0 und Mozilla Firefox ($\geq$ 47.0.1) ist der Geckodriver[11]. Dieser nimmt Befehle im WebDriver-Protokoll von Selenium entgegen und übersetzt sie in das Marionette-Protokoll für den Firefox Browser.

Überblick

Für die Automatisierung kommen folgende Anwendungen zum Einsatz:

- Mozilla Firefox 55.0.2 in unmodifizierter Form und ohne Erweiterung,
- Selenium 3.5.0. für die programmierbare Steuerung des Browsres und
- Geckodriver v16.01 für die Übersetzung zwischen Selenium und Browser (Marionette-Protokoll).

Unterschiede

Eine Verhaltensänderung des Browsers kann durch die Nutzung des Marionette-Protokolls mit Geckodrivers nicht ausgeschlossen werden. Allerdings handelt es sich um die am wenigsten invasive Lösung zur Umsetzung der Automatisierung. Selenium und Geckodriver sind eigenständige Prozesse, die in der Prozessüberwachung von den Ergebnissen des Browsers (Firefox) getrennt behandelt werden. Auch bei einer Nutzung von OpenWPM kommen diese Anwendungen zum Einsatz und werden durch die Browser-Erweiterung `openwpm.xpi` ergänzt.

8 https://www.seleniumhq.org/, abgerufen am 29.03.2018.
9 https://firefox-source-docs.mozilla.org/testing/marionette/marionette/index.html, abgerufen am 06.03.2018.
10 https://www.w3.org/TR/webdriver/, abgerufen am 06.03.2018.
11 https://github.com/mozilla/geckodriver/releases, abgerufen am 06.03.2018.

Messung

Messungen

Es ist zu hinterfragen, auf welche Weise sich zusätzliche Informationen aus dem Browser extrahieren lassen, ohne eine Modifikation des Browsers oder dessen Erweiterung durchzuführen. Um diese Frage zu klären, eignet sich eine Betrachtung der bereits integrierten Messwerkzeuge. Die Firefox-Werkzeuge für Web-Entwickler ermöglichen einen technischen Einblick in den Aufbau der Webseite (Inspektor) und in den Netzwerkverkehr. Diese Werkzeuge sind in Abbildung A.5 von Anhang A abgebildet. Im Reiter „Netzwerkanalyse“ sind alle Anfragen an den Webserver und dessen Antworten zu sehen.

HAR-Log

Ein Export dieser Anfragen und Antworten ist seit Mozilla Firefox Version 41 möglich. Dabei wird ein HAR-Log erstellt, welches das Speichern der Netzwerkkommunikation im JSON-Format erlaubt. Diese Log-Datei wird im Browserprofil gespeichert und kann nach Beendigung des Browsers weiterverarbeitet werden.

Aufbau eines HAR-Log

Aufbau

HAR (HTTP Archive[12]) ist ein Dateiformat, welches den Export und die Verarbeitung von Daten aus HTTP-Sitzungen ermöglicht. HAR-Logs können in allen gängigen Browsern (Google Chrome, Mozilla Firefox und Microsoft Edge) extrahiert werden. In Tabelle 10.1 wird eine Übersicht zu den enthaltenen Objekttypen geboten, die innerhalb der Spezifikation genauer definiert werden. Ein Beispiellog wurde in Kapitel 8 in Quelltext 8.1 vorgestellt.

Durchführung von Messungen

Durchführung

Die Durchführung einer Messung mit dem WebChecker umfasst die folgenden fünf Schritte:

1. Vorbereitung des Browserprofils.
2. Initialisierung des Browsers.
3. Aufruf der Webseiten.
4. Beendigung des Browsers.
5. Sicherung des Browserprofils.

Ablauf

In der Startphase (Vorbereitung und Initialisierung) wird das Browserprofil entsprechend der gewünschten Konfigurationen vorbereitet. Alternativ kann ein bereits vorhandenes Profil geladen und angepasst werden. Dies umfasst die notwendigen Anweisungen zur Erstellung eines HAR-Logs. Anschließend werden die zu prüfenden Webseiten aufgerufen. Für jede Webseite wird ein eigenes HAR-Log generiert. Die Beendigung des Browsers dient dem Abschluss aller Schreibprozesse auf dem System, damit das Browserprofil in einem konsistenten Zustand gesichert werden kann.

Kommunikation

Der WebChecker übermittelt den aktuellen Status über einen Socket

12 http://www.softwareishard.com/blog/har-12-spec/, abgerufen am 06.03.2018.

Objekttyp	Beschreibung
`log`	Metainformationen zum Log (z. B. Version)
`creator`	Applikation, die das Log erzeugt hat
`browser`	Informationen zum verwendeten Browser
`pages`	Geöffnete Seiten/Tabs
`pageTimings`	Laufzeitinformationen zu geöffneten Seiten/Tabs
`entries`	Eintrag, der auf jede Anfrage (request) und Antwort (response) verweist
`request`	Angeforderte Ressourcen aus dem Web
`response`	Antwort vom Webserver auf die angeforderten Ressourcen
`cookies`	Informationen zu Cookies
`headers`	Übersicht zu Request- und Responseheader
`queryString`	Informationen zu GET-Parametern bei Anfragen
`postData`	Übertragene POST-Anfragen
`params`	Zusätzliche POST-Parameter
`content`	Informationen zur Inhaltskodierung (z. B. mimeType)
`cache`	Zustand des Caches
`timings`	Laufzeitinformationen zu Ressourcen

Tabelle 10.1: Überblick zu Objekttypen im HAR-Log.

(UDP) zum DisTrack-Framework. Auf diese Weise kann der Status verfolgt und diese Kommunikationsverbindung ggf. für weitere Zwecke genutzt werden.

Weitere Funktionen

Weitere Funktionen

Neben der Kernaufgabe, die Netzwerkkommunikation des Browsers durch ein HAR-Log zu sichern, stellt der WebChecker weitere Funktionen zur Ausgestaltung von Analysen bereit:

PRIVATER MODUS. Der WebChecker ist in der Lage, den Browser im privaten Surfmodus zu starten.

WARTEZEITEN. Üblicherweise beenden Werkzeuge die Analyse, sobald die Webseite vollständig geladen wurde. Betrachtungszeiten von nur wenigen Millisekunden entsprechen jedoch nicht dem üblichen Nutzerverhalten. Aus diesem Grund lassen sich Mindestwartezeiten spezifizieren, die der Browser abwartet.

MANUELLER EINGRIFF. Das bereits vorgestellte WebDriver-Protokoll ermöglicht grundlegende Steuerungsmöglichkeiten. Dies umfasst allerdings nicht alle Interaktionen, die mit einem Browser möglich sind. Der WebChecker erlaubt daher, die Ausführung anzuhalten und diese Funktionen manuell auszuführen. Ein Beispiel ist die Löschung der Cookie- und Browserhistorie.

SCREENSHOT. Nach jedem Abruf einer Webseite wird ein Screenshot erstellt und mittels Base64-Kodierung dem Ergebnisbericht hinzugefügt.

10.5.2 *Methoden und Berichte*

Übersicht

Innerhalb der Entwurfsphase in Abschnitt 10.4.2 wurde eine Notation zur Abbildung von Tests eingeführt. Diese muss programmatisch umgesetzt werden. Der Test muss kodiert und durch das DisTrack-Framework verarbeitet werden können. Dies umfasst die Durchführung sowie die anschließende Auswertung der Daten in Form von Berichten.

Testklassen

Klassen

Mittels geeigneter Klassen sollen Tests, wie beispielsweise in Abbildung 10.2 dargestellt, modelliert und anschließend ausgeführt werden. Ein kompletter Test besteht aus mindestens einem Schritt oder mehreren Schritten. Für jeden Schritt ist ein System- und ein Profilzustand notwendig. Implementiert wird dies über eine Objektstruktur. Dabei werden Objekte von Klassen erstellt, um Test, Testschritte, System- und Profilzustand zu beschreiben und zu kombinieren. Zu diesem Zweck werden die folgenden Klassen definiert:

WTDANALYSIS Diese Klasse steht als Träger für den kompletten Test. Analyseschritte werden einem Objekt dieser Klasse über die `add_step()`-Methode hinzugefügt. Die Klasse stellt ebenfalls Methoden bereit, welche die notwendigen Steuerbefehle aus dem gesamten Modell erzeugt – dies wird in Abschnitt 10.5.3 näher beschrieben.

WTDANALYSISSTEP Jeder Schritt besteht aus den Öffnen einer Webseite und aus jeweils zwei System- und Profilzuständen: eines vor der Ausführung und eines nach der Ausführung. Dies ist nicht für alle Schritte disjunkt, denn der Systemstatus ist vor der Ausführung von 2 identisch mit dem Status nach der Ausführung von Schritt 1. Darüber hinaus können weitere Attribute gesetzt werden, beispielsweise Verzögerungen während der Laufzeit. Diese Zustände können über die Methoden `get_state_before()`, `get_state_after()`, `get_profile_before()` und `get_profile_after()` abgerufen werden.

WTDSTATE Objekte dieser Klasse repräsentieren den Status des Systems der virtuellen Maschine (Snapshot). Wesentlich ist hierbei, dass diese Objekte mit einem Zähler darüber verfügen, in wie vielen Tests sie eingesetzt werden. Wird ein solches Objekt beispielsweise nicht benutzt, kann ein Snapshot vollständig ausbleiben. Wird ein Snapshot erstellt, muss eine Löschung am Ende des Tests gewährleistet sein.

WTDPROFILE Wie der Systemzustand wird auch das Browserprofil als Klasse repräsentiert. Um nach Abschluss der Analyse auf das Ergebnisprofil zugreifen zu können, muss hier eine Zugriffsmöglichkeit auf die Profildaten bestehen. Auch bei Profilen wird ein Zähler für die Referenzen geführt, um nur tatsächlich benötigte Profile langfristig zu speichern.

Beispiel

Quelltext 10.1 zeigt ein Beispielablauf. Dabei wird das Modell aus Abbil-

dung 10.2 als Programm mit den beschriebenen Klassen kodiert.

Testaufbau

Der Test wird durch die Klasse `MyAnalysis` definiert, wobei der Name beliebig gewählt sein kann. Die Klasse muss allerdings von `Analysis` abgeleitet werden. Das DisTrack-Framework erzeugt ein Objekt dieser Klasse und ruft die `analysis()`-Methode aus Zeile 3 auf. Dabei wird ein Objekt der Klasse `WTDAnalysis` übergeben, das in Zeile 4 zur Erstellung eines neuen Schrittes genutzt wird. Dabei handelt es sich um ein neues Objekt der Klasse `WTDAnalysisStep`. Das daraus resultierende Browserprofil soll im zweiten Schritt genutzt werden. Deshalb wird in Zeile 5 ein Objekt angefordert, welches das resultierende Browserprofil nach Ausführung von Schritt 1 repräsentiert. Anzumerken ist, dass an dieser Stelle keine Webseite geöffnet wurde, da es sich um ein vereinfachtes Beispiel handelt. Andernfalls würde der Webseitenaufruf als Parameter dem `add_step()`-Aufruf übergeben. Die restlichen Zeilen folgen dem gleichen Prinzip. Aufrufe bewirken keine unmittelbare Ausführung, sondern konstruieren ausschließlich das Modell. Erst mit Aufruf der `start()`-Methode in Zeile 15 wird der Test ausgeführt, was detaillierter in Abschnitt 10.5.3 beschrieben wird.

```
1  class MyAnalysis(Analysis):
2
3          def analysis(self, analysis):
4              t1 = analysis.add_step()
5              p1 = t1.get_profile_after()
6
7              t2 = analysis.add_step(wtdprofile=p1)
8              p2 = t2.get_profile_after()
9              s2 = t2.get_state_after()
10
11             t3 = analysis.add_step(wtdprofile=p2, wtdstate=s2)
12             s3 = t3.get_state_after()
13
14             t4 = analysis.add_step(wtdprofile=p2, wtdstate=s3)
15             analysis.start()
```

Quelltext 10.1: Beispielmodell aus Abbildung 10.2 als Programm kodiert.

Berichte

Ziele

So wie neue Tests durch Objekte modelliert werden können, kann man auf die Messergebnisse durch entsprechende Methoden der Objekte zugreifen. Zu diesem Zweck verfügen die Klassen `WTDAnalysisStep` und `WTDProfile` über Zugriffsmethoden. In Quelltext 10.2 wird ein einfaches Verarbeitungsbeispiel vorgestellt. Die Klasse `MyReport` wird von `Report` abgeleitet und nach Abschluss der Ausführung aufgerufen. Der Methode `report` wird ein Objekt der Klasse `WTDAnalysis` übergeben, wie es in Quelltext 10.1 aufgebaut ist. Dies kann in Zeile 4 in die einzelnen Schritte zerlegt werden. Die Methode `get_task_report` ermöglicht den Zugriff auf den JSON-formatierten Ergebnisbericht (vgl. Abschnitt 9.4.4). Über das `behavior`-Attribut kann explizit auf den Abschnitt der Verhaltensdaten im Ergebnisbericht zugegriffen werden. Die Methode `get_processes_sum-`

mary() führt eine Strukturierung durch und filtert vorab nicht benötigte Daten.

```
1  class MyReport(Report):
2
3          def report(self, analysis):
4          t1,t2,t3,t4 = analysis.get_steps()
5
6          tr1 = self.get_task_report(t1)
7          sum1 = tr1.behavior.get_processes_summary()
8
9          tr3 = self.get_task_report(t3)
10         sum2 = tr3.behavior.get_processes_summary()
```

Quelltext 10.2: Verarbeitungsbeispiel eines Testergebnisses von DisTrack.

10.5.3 *Ablaufsteuerung*

Steuerung

Um modellierten Tests durch das DisTrack-Framework auszuführen, werden Teile der Sandbox durch Klassen abgebildet und stellen die Kommunikation zur Sandbox her. Somit ist die Übergabe von neuen Anweisungen sowie die Überwachung bestehender Aufgaben über diese Schnittstelle möglich. Infolgedessen können neue Analyseaufträge auf Basis des spezifizierten Modells generiert und übertragen werden.

Cuckoo Abstraktionen

Klassen

Zum Steuern des Ablaufs muss DisTrack mit der Sandbox (Cuckoo) interagieren. Zu diesem Zweck wurden Klassen geschaffen, die einen Zugriff auf die Sandbox vereinfachen und von der genauen Umsetzung der Kommunikation abstrahieren.

CUCKOO Ein Objekt dieser Klasse stellt den Zugriff zur Cuckoo Sandbox her. So können über Methoden wie `get_task_view()` alle bisherigen Analysen der Sandbox ausgelesen oder über die `submit`-Methode neue Analyseanfragen gestellt werden.

CUCKOOSUBMITREQUEST Diese Klasse verkapselt neue Analyseanforderungen an die Sandbox. Sie beinhaltet insbesondere eine Liste der Optionen und ermöglicht, aus den hinterlegten Daten einen POST-Request für die REST-API der Sandbox zu erstellen.

CUCKOOWEBCSUBMITREQUEST Diese Klasse erweitert die `CuckooSubmitRequest`-Klasse um spezielle Attribute für den WebChecker (vgl. Abschnitt 10.6.1).

CUCKOOTASK Ein Task ist der in Abschnitt 9.4.1 beschriebene Analysedurchlauf der Sandbox. Dieser kann in verschiedenen Zuständen sein: Vorbereitung, in Ausführung, Verarbeitung und Berichtigung. Sobald die Erstellung des Berichts abgeschlossen ist, gilt der Auftrag als vollständig verarbeitet.

CUCKOOWEBCTASK Diese Klasse erweitert den CuckooTask um spezielle Attribute des WebCheckers (vgl. Abschnitt 10.6.1).

Auf Basis dieser Abstraktionen ist eine Interaktion mit der Sandbox möglich. Im nächsten Schritt wird beschrieben, wie aus den Modellen die Ansteuerung der Sandbox realisiert wird. Die Umsetzung dieser Ansteuerung zeigt Quelltext 10.1.

Generierung der Sandbox-Optionen

Optionen

Die Klasse `CuckooWebcSubmitRequest` wird von `CuckooSubmitRequest` abgeleitet und ermöglicht eine Übersetzung des Modells in die Sandbox-Optionen. Wird beispielsweise die Methode `set_sourcesnapshot()` aufgerufen, wird dem Analyseauftrag die Option `source_snapshot` hinzugefügt. Bei Absendung des Analyseauftrags (`submit()`) wird diese Option als Parameter der Sandbox übergeben.

```
1 webc_targetprofile=3e180b4aaeb4ed9f.zip,webc_testname=#1
2 additional_files=/home/cuckoo/CWD/storage/analyses/1/webc/3e180b4
      aaeb4ed9f.zip,target_snapshot=04021cd7904ecb67,webc_
      sourceprofile=3e180b4aaeb4ed9f.zip,webc_targetprofile=304e735
      d10fccc76.zip,webc_testname=#2
3 additional_files=/home/cuckoo/CWD/storage/analyses/2/webc/304e735
      d10fccc76.zip,source_snapshot=04021cd7904ecb67,target_
      snapshot=a3d297743cd64b5d,webc_sourceprofile=304e735d10fccc
      76.zip,webc_testname=#3
4 additional_files=/home/cuckoo/CWD/storage/analyses/2/webc/304e735
      d10fccc76.zip,delete_snapshot=04021cd7904ecb67;a3d297743cd64b
      5d,source_snapshot=a3d297743cd64b5d,webc_sourceprofile=304e
      735d10fccc76.zip,webc_testname=#4
```

Quelltext 10.3: Optionen, die zur Abbildung des Modells von Quelltext 10.1 der Sandbox übergeben werden.

Beispiel

In Quelltext 10.3 können die aus Quelltext 10.1 erzeugten Optionen eingesehen werden. Jede Zeile entspricht einem Analyseschritt. Einige der Parameter sind bereits Bestandteil der Sandbox, andere werden erst durch Erweiterungen ermöglicht, wie sie in Abschnitt 10.6 noch beschrieben werden.

Aufbau der Optionen

Der erste Testschritt wird durch die Zeile 1 in Quelltext 10.3 abgebildet. Mit `webc_targetprofile` wird eine Sicherung des Browserprofils bewirkt. In Schritt 2, dessen Optionen in Zeile 2 zu sehen sind, wird dieses Profil als Quelle `webc_sourceprofile` ausgewählt und mittels `additional_files` an die Sandbox übertragen. Schritt 2 sieht ebenfalls die Sicherung des Systemzustands vor, was durch die Option `target_snapshot` erreicht wird. In Schritt 3 wird dieser Systemzustand verwendet, weshalb in Zeile 3 die Option `source_snapshot` gesetzt ist. In Zeile 4 wird die Löschung von Snapshots mittels `delete_snapshot` befehligt.

Ablaufüberwachung

Ablauf

Das Abarbeiten der Analyseanforderungen erfolgt schrittweise: Erst wenn ein Analyseschritt vollständig abgeschlossen ist, wird der nächste gestartet.

Auf diese Weise ist sichergestellt, dass die Sandbox nicht überlastet wird. Um diese Überwachung durchzuführen, ermöglicht die `Cuckoo`-Klasse über die `wait_for_report_by_request()`-Methode eine Unterbrechung der Ausführung, bis der Auftrag abgearbeitet wurde. Durch aktives Warten wird alle drei Sekunden eine Prüfung durchgeführt, ob der Auftrag vollständig ausgeführt wurde.

10.5.4 *Datenspeicherung*

Speicherung

Bei Nutzung des DisTrack-Frameworks fallen viele Mess- und Analysedaten an. Dabei kommt eine Überführung dieser Daten in eine gemeinsame Datenbank in Betracht. Um der doppelten Datenhaltung zu entgehen, wurde auf eine weitere Speicherung verzichtet. Die beschriebenen Klassen ermöglichen einen transparenten Zugriff auf alle Messergebnisse unabhängig von ihrem Speicherort. Da diese Daten zur Verarbeitung in den Arbeitsspeicher geladen werden, ist keine Reduktion der Verarbeitungsgeschwindigkeit zu erwarten.

Speicherform

Durchgeführte Tests können innerhalb des DisTrack-Frameworks gespeichert werden. Dabei wird die Objektstruktur, wie sie in Abschnitt 10.5.2 beschrieben wurde, in das Dateisystem abgelegt. Mittels `analysis.load_config()` können diese geladen und verwendet werden. Auf diese Weise ist ein gegenseitiger Vergleich der Ergebnisse verschiedener Tests möglich.

10.6 Modifikation an der Sandbox

Cuckoo Erweiterung

Die Erweiterung der Cuckoo Sandbox zu speziellen Analysezwecken ist nicht unüblich. Ferrand [54] führt Modifikationen durch, um zu verhindern, dass die Schadsoftware die Ausführung in einer Analyseumgebung erkennt und das Verhalten ändert. Provataki et al. [147] modifizieren die Cuckoo Sandbox um bessere Vergleichsmöglichkeiten von Analyseergebnissen untereinander.

Übersicht

Zur Verbesserung der Analysemöglichkeiten wurden Modifikationen an der Sandbox vorgenommen, die in den folgenden Unterabschnitten beschrieben werden. Eine Beschreibung der Komponenten wurde bereits in Abschnitt 9.4.1 vorgenommen, die im Folgenden modifiziert werden.

10.6.1 *WebChecker Analyse- und Verarbeitungsmodul*

Module

Für den in Abschnitt 10.5.1 beschriebenen WebChecker muss für die Ausführung innerhalb der Sandbox ein neues Analyse- sowie ein neues Verarbeitungsmodul erstellt werden. Wird anstelle des WebCheckers ein anderes Analysewerkzeug eingesetzt, müssen entweder die Ergebnisdaten konvertiert werden oder die Sandbox-Module an das neue Ergebnisformat angepasst werden.

Analysemodul

Das Analysemodul führt die WebChecker-Applikation aus und übergibt dafür notwendige Parameter. Alle Analysemodule der Sandbox befinden sich im Verzeichnis `analyzer/windows/modules/packages`. Für das neue webc-Modul wird an dieser Stelle eine neue Datei `webc.py` angelegt. Der Programmcode kann in Quelltext C.3 in Anhang C eingesehen werden. Es dient der Übersetzung und Weiterleitung von Optionen an den WebChecker, die an die Sandbox gerichtet sind. Folgende Parameter können dem Analysemodul übergeben werden:

WEBC_TESTNAME Ein Name/Bezeichner für den Testdurchlauf, um die Zuordnung zu erleichtern.

WEBC_SOURCEPROFILE Pfad zum Browserprofil, dass für die Analyse genutzt werden soll. Ist dieses Attribut nicht gesetzt, wird ein neues Profil angelegt.

WEBC_TARGETPROFILE Zielpfad für das entstandene Browserprofil nach der Ausführung. Ist dieses Argument nicht gesetzt, wird das temporäre Profil verworfen.

WEBC_TIMEOUT Maximale Laufzeit bis die Analyse abgebrochen wird. Der Standardwert ist 120 Sekunden.

WEBC_WAITTIME Zeitspanne, die der Browser geöffnet bleiben soll. Sofern nicht angegeben, wird der Browser unmittelbar nach Abschluss des Ladeprozesses geschlossen.

WEBC_PRIVATE Öffnen des Browsers im privaten Modus.

WEBC_WAITAFTER Möglichkeit der Interaktion nach dem Öffnen des Browsers und vor dem Besuch der Webseite.

WEBC_WAITBEFORE Eine weitere Interaktionsmöglichkeit vor dem Schließen des Browsers.

HAR-Log

Dem Analysemodul wird ebenfalls ein Pfad für die Ablage der Ergebnisse (HAR-Log, Browserprofil, etc.) angegeben. Nach Abschluss des Analysevorgangs werden die Ergebnisdaten vom Gast an den Host übertragen. Für die weitere Verarbeitung ist das WebChecker-Verarbeitungsmodul verantwortlich. Zu diesem Zweck wurde im Verzeichnis `processing/` die Datei `WebChecker.py` erstellt, deren Inhalt in Quelltext C.4 in Anhang C einsehbar ist.

Ergebnisse

In Abschnitt 9.4.4 wurde bereits der Aufbau einer Cuckoo Ergebnisdatei im JSON-Format beschrieben. Das Verarbeitungsmodul fügt einen Schlüssel `webchecker` dem Bericht hinzu und ergänzt diesen um die erhobenen HAR-Logs, die sich bereits im JSON-Format befinden. In Quelltext 10.4 wird ein Beispieleintrag abgebildet. Dabei wird an der Stelle HARLOG das Logergebnis eingefügt, das bei Abruf von URL erzeugt und über den Eintrag HARLOGS in der Ergebnisdatei gelistet wird. Ein Screenshot wird Base64-kodiert an der Stelle SCREENSHOT hinterlegt.

```
"webchecker": {
    "exportfiles": [],
    "targetprofile": "022a76dc50d624.zip",
    "timeout": 120,
    "sourceprofile": "C:\\tmpmxcrmi\\files\\6dac0264d8d3f.zip",
    "date": "2017-09-25 20:30:43.004000",
    "har": {},
    "outputfolder": "C:\\Users\\lab\\AppData\\Local\\Temp\\
        webc_results",
    "checks": {
        "1": {
            "screenshot_data": SCREENSHOT,
            "screenshot": "screenshot.png",
            "url": "URL",
            "exported": [],
            "time": 94,
            "har": [
                HARLOG
            ],
            "harfiles": [
                HARLOGFILES
            ],
            "result": true
        }
    }
},
```

Quelltext 10.4: Eintrag für die WebChecker-Analyse im Ergebnisbericht der Sandbox.

10.6.2 *Dumptls*

Problemstellung

Ein grundsätzliches Problem bei Analyse des Netzwerkverkehrs ist der Einsatz von Transportsicherheitsmaßnahmen – dies trifft insbesondere auf die Protokolle SSL/TLS zu. Es findet eine Ende-zu-Ende-Verschlüsselung der Daten statt, die beim Browser des Clients beginnt und erst am Webserver auf dem Zielserver endet. Damit wird offensichtlich, dass die Sandbox, die zwischen dieser Übertragung steht, nur bedingt Zugriff auf die Inhaltsdaten erhält.

Möglichkeiten

Grundsätzlich kann dieses Problem durch den Einsatz eines Proxys umgangen werden, der zu einem Aufbrechen der verschlüsselten Verbindung in der Lage ist. Dies kann u. a. durch mitmproxy[13] erreicht werden. Dabei zeigen sich jedoch verschiedene Probleme.

Nachteile

Damit ein Man-in-the-Middle-Angriff, wie er hier unternommen wird, funktionieren kann, muss der Proxy für die angeforderten Webseiten ein gültiges Zertifikat vorlegen können. Umgesetzt wird dies üblicherweise durch ein eigenes Wurzelzertifikat, das vom Client als gültiges angenommen wird. Der Proxy erstellt dann für jede geforderte Webseite ein neues Zertifikat, das vom eigenen Wurzelzertifikat signiert wurde. Dies bedeutet jedoch, dass für jede über HTTPS erreichbare Webseite ein anderes Zertifikat als das tatsächliche zum Einsatz kommt. Dies kann insbesondere bei Einsatz

13 https://mitmproxy.org/, abgerufen am 24.02.2018.

von Certificate Pinning (RFC 7464 [51]) zu Problemen mit der Akzeptanz des Zertifikates führen. Bei zusätzlichen Absicherungen über die DNS-Infrastruktur[14] würde dieser Ansatz ebenfalls fehlschlagen.

Alternative

Der Einsatz einer solchen Technik stellt einen schwer abschätzbaren Eingriff in die Kommunikation dar. Für DisTrack wurde sich für einen weniger invasiven Weg entschieden. Dabei wird der Mozilla Firefox Browser durch die Umgebungsvariable SSLKEYLOGFILE dazu angewiesen, die SSL/TLS Mastersecrets in eine Logdatei zu schreiben. Unter Verwendung dieser Mastersecrets können verschlüsselte Verbindungen auch dann nachträglich entschlüsselt werden, wenn die Verbindung mittels Perfect Forward Secrecy geschützt wird (z. B. ein Diffie-Hellman-Schlüsselaustausch). Die Daten werden im NSS Keylog Format[15] bereitgestellt und sind beispielhaft in Quelltext 10.5 abgebildet. Bei diesen drei Einträgen pro Zeile handelt es sich um einen Bezeichner CLIENT_RANDOM, die Nonce des Clients das Mastersecret. Aus dem Mastersecret leiten sich alle Schlüssel ab, die innerhalb der SSL/TLS-Sitzung eingesetzt werden.

```
1 # SSL/TLS secrets log file, generated by NSS
2 CLIENT_RANDOM b598c33759aaa42cd834d296856c3e76d640ed74cb42c7d3fec
      5fe7a4a9e4062 8198be63e9013342374be0c1dc63dba9f11048f9a02d25
      cc07dea6b29d63b3cc5c139f834efe93b6110c2010c88952b6
3 CLIENT_RANDOM 54e015cf30e355c517acfb40ae30b275aad2b17dce644dc13c1
      a17cd51da6805 5fecbd35dabf1b32b334d44130e25ebafe669901f031af
      34b1c32d4b87f30fcfb7f963f1677fbd7794fc672e00478c33
4 CLIENT_RANDOM fb691cc8107d69f951d0902b863b8e183ebdf465c05a685719f
      9669787b5416d 3d9bc83155a12e75525aacd67ea1f54833a934471bab4eb
      856b864dd151f9a761208f1312a677a27dbf25d2448e80edc
5 CLIENT_RANDOM fd754f48bcd8291b9b27b0c020eed48059fc3e564dfc3cab2b
      80c946c6841919 ae8fae22a0541b4fa10dde9893ca1963e0f03ef557c120
      ed92332161ca89fbfcde5a75c2a6801b1716bb3ec50a0322cb
6 ...
```

Quelltext 10.5: Beispiele der SSLKEYLOGFILE-Datei.

Umsetzung

Die Cuckoo Sandbox enthält bereits eine Komponente, die eine Entschlüsselung der Netzwerkkommunikation ermöglicht. Diese beschränkt sich auf das Schlüsselmaterial, das durch Infiltration des LSASS-Prozesses erhoben wurde[16]. Auf diese Weise wird Schlüsselmaterial von Verbindungen gesammelt, die durch den Internet Explorer Browser initiiert werden. Dies schließt allerdings nicht Verbindungen des Mozilla Firefox Browsers mit ein. Um dies auf den Firefox Browser auszuweiten, sind Modifikationen an der Sandbox zur Speicherung und Verarbeitung der Daten notwendig.

Integration

Sofern die Umgebungsvariable gesetzt wurde, findet eine automatische Speicherung der Secrets unter dem gewählten Verzeichnispfad statt. Nach Beendigung des Browsers muss eine Übertragung der Logdatei vom Gast-

14 DNSKEY Resource Record für TLS nach RFC 4034 [154].

15 https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/Key_Log_Format, abgerufen am 24.02.2018.

16 https://github.com/cuckoosandbox/cuckoo/blob/984e5258ccdf196306df37b248fe6ea1ef4c890b/cuckoo/data/analyzer/windows/modules/auxiliary/dumptls.py, abgerufen am 25.02.2018.

an das Hostsystem erfolgen. Zu diesem Zweck wird die bereits existierende `dumptls.py` in `analyzer/windows/modules/auxiliary/` erweitert, wie in Quelltext 10.6 zu sehen ist.

```
1
2  def stop(self):
3    upload_to_host(os.environ['SSLKEYLOGFILE'], os.path.join("
         network","SSLKEYLOGFILE.txt"))
```

Quelltext 10.6: Modifikation der `dumptls.py` zur Übertragung der SSLKEYLOGFILE-Datei.

Modifikation

Zur Verarbeitung der Ergebnisse wurde die `get_tlsmaster`-Methode aus `/processing/network.py` so modifiziert, dass das SSLKEYLOGFILE zusätzlich zum bestehenden Schlüsselmaterial verarbeitet wird. Das Ergebnis der Modifikation kann in Quelltext C.1 in Anhang C eingesehen werden.

Ergebnis

Auf diese Weise wird eine Übertragung und die anschließende Entschlüsselung umgesetzt. Die verwendeten Bibliotheken zur Entschlüsselung müssen einen aktuellen Stand aufweisen. Schlägt der Vorgang fehl, wird nicht der gesamte Verarbeitungsprozess abgebrochen, sondern lediglich die jeweilige Verbindung nicht ins Ergebnis aufgenommen.

10.6.3 *Additional Files*

Weitere Dateien

Wird in der Cuckoo Sandbox eine neue Analyse gestartet, wird der Analyzer über den Agent vom Host- auf das Gastsystem übertragen. Die Übermittlung weiterer Daten ist in der Cuckoo Implementierung nicht vorgesehen. Wie in Abschnitt 10.5.3 bereits erwähnt wurde, ist dies insbesondere dann notwendig, wenn eigene bzw. bestehende Browserprofile verwendet werden sollen. Aus diesem Grund wurde die Option `additional_files` eingeführt, die auf eine Datei oder einen Ordner zeigen kann.

Umsetzung

Erreicht wird dies durch eine Erweiterung der `analyzer_zipfile`-Funktion in `core/guest.py`:

```
1
2  if additional_files:
3      if os.path.isdir(additional_files):
4          for root, dirs, files in os.walk(additional_files):
5              for name in files:
6                  zip_file.write(os.path.join(root,name), os.path.
                       join("files",name))
7      else:
8          zip_file.write(additional_files, os.path.join("files",os.
               path.basename(additional_files)))
```

Quelltext 10.7: Erweiterung der `core/guest.py` zur Übertragung weiterer Dateien vor der Analyse.

Anschließend kann die Sandbox beispielsweise mit der Kommandozeile:

```
1  cuckoo --cwd=/home/cuckoo/CWD submit --package webc -u example.
       com -options additional_files=/home/cuckoo/furtherstuff
```

aufgerufen werden, was neben der Auswahl des webc-Analysemoduls auch zur Übertragung der Dateien im Verzeichnis /home/cuckoo/furtherstuff führt. Die angegebenen Dateien stehen während der Analyse zur Verfügung.

10.6.4 *Snapshot-Erweiterung*

Steuerung der VM

Ein weitere benötigte Funktionalität ist die Steuerung von Snapshots der virtuellen Umgebung. So soll die Möglichkeit geschaffen werden, eine Analyse von einem spezifischen Sicherungspunkt zu starten und ggf. einen neuen nach der Analyse zu erstellen. Ebenfalls sollen bestehende Snapshots löschbar sein, sofern diese nicht mehr benötigt werden. Aus diesem Grund werden die folgenden Optionen eingeführt:

- source_snapshot,
- target_snapshot und
- delete_snapshot.

Nutzung

Bei Start der virtuellen Umgebung wird der Snapshot ausgewählt, der mittels der source_snapshot-Option angegeben wird. Die Methode prestop wird vor Beendigung der virtuellen Umgebung ausgeführt und sorgt für eine Sicherung des in target_snapshot angegebenen Zustands. Des Weiteren löscht prestop die nicht länger benötigten Snapshots, sofern diese über die delete_snapshot-Option genannt sind. Die notwendigen Erweiterungen können in Quelltext C.2 in Anhang C eingesehen werden.

10.7 Anpassungen für OpenWPM

Alternative

Anstelle des WebCheckers ist der Einsatz von OpenWPM[17] zur Messung möglich. Dabei ist zu berücksichtigen, dass OpenWPM für Linux-basierte Systeme entworfen wurde. Aus diesem Grund ist eine manuelle Installation und Konfiguration für den Betrieb unter Windows 7 notwendig. Im Zuge der Prüfung wurden diese Änderungen durchgeführt und hier zusammengefasst:

- Installation von Python und benötigten Bibliotheken[18],
- Installation von Firefox in der Version 52 (ESR),
- Installation des Geckodrivers in der letzten Version (v0.20.0),
- Startscript zur Entgegennahme der WebC-Startparameter; vgl. Quelltext C.3.

17 https://github.com/citp/OpenWPM/tree/e2e7dcdfc6b18dcbf73196fffd6a00d610d34a13, abgerufen am 29.03.2018.

18 https://github.com/citp/OpenWPM/blob/e2e7dcdfc6b18dcbf73196fffd6a00d610d34a13/requirements.txt, abgerufen am 29.03.2018.

- Optional: Überführung der Ergebnisse der sqlite-Datenbank nach JSON.

10.8 VERSIONSINFORMATIONEN

Hostsystem

Die Entwicklung und alle Evaluationen wurden auf Basis der Cuckoo Sandbox in der Version 2.0.3 vom 19.05.2017 durchgeführt. Installiert wurde diese auf einem Ubuntu 16.04.3 LTS (xenial) mit einem Updatestand vom 21.08.2017. Die Programmiersprache ist Python in der Version 2.7.12. Die Versionsstände der verwendeten Python-Bibliotheken können in Anhang E.2 eingesehen werden. Die eingesetzte Virtualisierungssoftware ist VMWare 12.5.9.

Gastsystem

Im Gastsystem kommt Windows 7 (32-Bit) zum Einsatz, mit einem Updatestand vom 21.08.2017. Der verwendete Browser ist der Mozilla Firefox 55.0.2 (32-Bit, englisch) in Verbindung mit dem Geckodriver v16.01 und Selenium 3.5.0. Zusätzlich installiert wurde Python 2.7.12 (wie oben), Microsoft .NET Framework 4.6.1, Java 8 Update 144 vom 21.08.2017. Die automatische Installation von Updates wurde deaktiviert, um den damit verbundenen Netzwerkverkehr zu unterdrücken.

11 EVALUATION VON DISTRACK

Zusammenfassung: In diesem Kapitel wird eine Evaluation des Frameworks und seiner Komponenten durchgeführt. Die Evaluation umfasst Tests des Werkzeugs, um die Funktionsweise sicherzustellen und eine tiefere Betrachtung der erzeugten Messdaten. Durch die Entwicklung, Ausführung und Auswertung von Analysemodellen werden die Besonderheiten des Frameworks hervorgehoben.

11.1 EINLEITUNG

Motivation

Mit Forschungsfrage RQ-4 wurde nach einem Mehrwert der verhaltensbasierten Analyse von Web-Tracking außerhalb des Browsers gefragt. Bislang wurden keine Arbeiten zu dieser Fragestellung durchgeführt, wie Kapitel 9 aufgezeigt hat. In Kapitel 10 wurde das Rahmenwerk einer solchen Analyseumgebung entworfen und implementiert. Somit werden Sandbox-gestützte Analysen von Webseiten ermöglicht. Zu diesem Zweck wurde prototypisch der WebChecker implementiert, welcher browsereigene Funktionen zur Datenerhebung nutzt und somit die am wenigsten invasive Form der Umsetzung ist. Wie andere Werkzeuge in DisTrack eingesetzt werden können, wurde in Abschnitt 10.7 beschrieben.

Methodiken

Es stellt sich die Frage, auf welche Weise eine Evaluation des Frameworks durchgeführt werden kann. Zu diesem Zweck werden verwandte Arbeiten näher betrachtet, in denen ein Werkzeug oder ein Framework implementiert wird. Ziel ist, die dort durchgeführten Methodiken auf diesen Anwendungsfall zu adaptieren.

- Acar et al. [1] entwickeln mit FPDetective ein Werkzeug zur Erkennung von Fingerprinting-Verfahren. Für die Evaluierung wird diese Erweiterung eingesetzt, um eine Schwachstelle im Tor-Browser zu finden, die für Fingerprint-basiertes Web-Tracking ausgenutzt werden kann.
- Libert [107] führt keine gesonderte Werkzeugevaluierung von WebX-Ray durch.
- Das Werkzeug Adfisher wird von Datta et al. [40, 39] ausführlich beschrieben. Sie führen keine detaillierte und strukturierte Evaluierung durch.
- Mayer und Mitchell [115] stellen für FourthParty drei Designprinzipien auf.
- Von Englehardt und Narayanan [48] werden zur Evaluierung von OpenWPM vier Anforderungen formuliert, die in der Werkzeugimplementierung berücksichtigt werden müssen.

Anforderungen

Die von Englehardt und Narayanan [48] definierten Anforderungen für OpenWPM[1] wurden in Abschnitt 10.3 vorgestellt. Darüber hinaus werden von Englehardt und Narayanan keine weiteren Werkzeugprüfungen oder Evaluationstechniken beschrieben. Die anforderungsbasierte Werkzeugprüfung ist somit Teil der vorliegenden Arbeit. Zur Prüfung der Anforderungen werden in Abschnitt 11.2 Werkzeugtests durchgeführt, die auf die Stabilität, Quantität, Korrektheit und Geschwindigkeit abzielen. Die Frage zur Qualität der Daten wird in Abschnitt 11.3 gesondert behandelt.

Werkzeugtests

Die Tests aus Abschnitt 11.2 und 11.3 beziehen sich auf DisTrack in Verbindung mit dem WebChecker (vgl. Abschnitt 10.5.1). Wie beschrieben wurde, kann anstelle des WebCheckers auch ein anderes Werkzeug in DisTrack eingesetzt werden. In diesen Abschnitten findet eine Prüfung dieser browsereigenen Analyseform unter Verwendung des WebCheckers statt, da dieser

- in Studie B der vorliegenden Arbeit (Abschnitt 8.2) verwendet wurde und
- in zukünftige Analysen diese Messmethode ohne weitere Evaluation einsetzen können.

Alleinstellungsmerkmale

So wie Acar et al. [1] das entwickelte Werkzeug zur Auffindung von Schwachstellen im Tor-Browser verwenden, soll auch in dieser Evaluation das DisTrack-Framework zur Bestimmung spezieller Tracking-Verfahren genutzt werden. Auf diese Weise wird der Mehrwert des Werkzeugs bewiesen. Diese Methodik soll in der vorliegenden Arbeit verwendet und Fälle aufgezeigt werden, in denen die Vorteile der Sandbox zur Durchführung der Analyse beitragen. In Abschnitt 11.4 werden anhand konkreter Analysemodelle die Vorteile des DisTrack-Frameworks demonstriert. Es werden Messverfahren vorgestellt, welche die Vorteile des Frameworks nutzen und mit alternativen Messverfahren nicht oder nur schwer zu messen sind. Auf diese Weise werden die Vorteile erkennbar.

Kapitelübersicht

Zukünftige Modelle und Einsatzmöglichkeiten werden in Abschnitt 11.5 beschrieben. Eine Vergleichsanalyse der Invasivität des Mozilla Firefox findet in Abschnitt 11.6 statt. Nach der Überprüfung der Anforderungen in Abschnitt 11.7 bilden eine Schlussfolgerung und ein Ausblick auf weitere Arbeiten (Abschnitt 11.8) den Abschluss des Kapitels.

11.2 WERKZEUGTESTS

Überblick

Im Folgenden werden Tests des Werkzeugs durchgeführt. Diese umfassen die Prüfung der Stabilität, eine Übersicht der erzeugten Menge an Messdaten, einen Vergleich mit einem anderen Browser und die Messung der Analysegeschwindigkeit.

1 Da OpenWPM eine Weiterentwicklung von FourthParty ist, werden die Anforderungen von Mayer und Mitchell berücksichtigt.

11.2.1 *Stabilität und Quantität*

Methodik

Für diesen Test wurden zehn aus den Top 50 der populärsten Webseiten sequenziell mittels der Sandbox gemessen. Diese Testmenge wurde bereits in Abschnitt 6.3.2 verwendet. In Tabelle 11.1 sind die Ergebnisse der Messung abgebildet. Die Spalte „Größe Report (kB)“ gibt die Größe des JSON formatierten Berichts an, wie er in Abschnitt 9.4.4 beschrieben wurde. Geladene/Modifizierte Dateien auf dem System sind darin nicht enthalten. Die Spalte „Größe WebC (kB)“ beschreibt die Größe des vom Browser extrahierten HAR-Logs; vgl. Abschnitt 10.5.1. Die letzte Spalte gibt an, ob die Analyse erfolgreich war oder Fehler aufgetreten sind.

Nr	Webseite	Zeitpunkt	Größe Report (kB)	Größe WebC (kB)	Erfolgreich
1	google.com	08.03.2018 13:39	130 205	1 502	✓
2	reddit.com	08.03.2018 13:43	269 903	5 683	✓
3	ebay.com	08.03.2018 13:47	180 224	5 146	✓
4	msn.com	08.03.2018 13:51	148 905	2 150	✓
5	washingtonpost.com	08.03.2018 13:55	394 205	11 735	✓
6	bbc.com	08.03.2018 13:59	461 816	7 694	✓
7	spiegel.de	08.03.2018 14:06	455 292	11 372	✓
8	yahoo.com	08.03.2018 14:12	195 679	4 426	✓
9	wordpress.com	08.03.2018 14:16	180 822	1 683	✓
10	nytimes.com	08.03.2018 14:21	509 677	13 174	✓

Tabelle 11.1: Sequenzielle Durchführung von Analysedurchläufen zur Stabilitätsprüfung.

Ergebnisse

Grundsätzlich ist die Sandbox in der Lage, gleichzeitig eine neue Messung durchzuführen und die Ergebnisse der vorherigen zu analysiert. Eine Vorabprüfung zeigte allerdings, dass dies zu einer Beeinträchtigung der Messgeschwindigkeit und auf diese Weise zu fehlerhaften Daten führen kann. Aus diesem Grund ist angeraten, erst nach vollständigem Abschluss eines Analysedurchlaufs eine neue Messung zu starten.

11.2.2 *Vergleich mit Google Chrome Browser*

Methodik

Um die korrekte Arbeitsweise der HAR-log basierten Analyse durch den WebChecker sicherzustellen, wird eine Vergleichsmessung mit dem Google Chrome Browser in der Version 65.0.3325.146 durchgeführt. Durch Öffnen der Entwicklertools dieses Browsers können die Netzwerkverbindungen bei Abruf einer Webseite genauer betrachtet werden. Grundsätzlich zeigt sich ein solcher Vergleich als schwierig, da Inhalte von Webseiten je nach Zeitpunkt der Abfrage abweichen können. Ebenfalls ist es möglich, dass sich die Inhalte je nach verwendetem User-Agent unterscheiden[2]. Infolgedessen wurden für diesen Test für Webseiten ausgewählt, die keine Werbeeinblen-

2 Der hier verwendete User-Agent entspricht den Angaben in Abschnitt 10.8.

dungen vornehmen. Auf diesen Webauftritten sind solche Abweichungen nicht zu erwarten.

Vergleich

In Tabelle 11.2 werden die Ergebnisse der manuellen Prüfung dargestellt. Der Zeitpunkt A ist die Abfrage der Webseite durch den WebChecker innerhalb der Sandbox, Zeitpunkt B ist die Abfrage durch den Google Chrome Browser. Der Vergleich umfasst nicht nur die Anzahl, sondern auch den Inhalt der abgerufenen Ressourcen, wobei aus Gründen der Übersichtlichkeit nur die Anzahl dargestellt wird. Die Zuordnung von A und B ist hier analog. Die letzte Spalte gibt Auskunft darüber, ob die abgefragten Ressourcen identisch waren.

Abweichungen

Die Differenz zwischen 0 und 3 ergibt sich aus der fehlenden Auflistung der Favicons. Dabei handelt es sich um eine Bilddatei, welche in der Titelleiste des Browsers nach Abruf der Webseite angezeigt wird. Der Abruf dieser Ressource wird weder im HAR-Log noch innerhalb der Entwicklerwerkzeuge (wie sie in Abbildung A.5 zu sehen sind) des Mozilla Firefox angezeigt. Das Favicon kann in den unterschiedlichen Bildgrößen 8x8, 16x16 und 32x32 Pixel nachgeladen werden, wodurch sich die Differenz von 0 bis 3 Ressourcen erklärt.

Ergebnis

Abgesehen von der beschriebenen Ausnahme zeigt sich eine Übereinstimmung der Messergebnisse wie in Tabelle 11.2 zu sehen ist.

Nr	Webseite	Zeitpunkt A	Zeitpunkt B	Anzahl A	Anzahl B	Bestanden
1	hochschule-trier.de	09.03.2018 20:32	09.03.2018 20:33	42	44	✓*
2	wikipedia.org	09.03.2018 20:35	09.03.2018 20:35	7	8	✓*
3	arbeitsagentur.de	09.03.2018 20:38	09.03.2018 20:39	29	32	✓*
4	blog.fefe.de	09.03.2018 20:51	09.03.2018 20:51	1	2	✓*
5	netzpolitik.org	09.03.2018 20:54	09.03.2018 20:55	60	61	✓*
6	kit.edu	09.03.2018 21:02	09.03.2018 21:02	55	56	✓*
7	www.kernel.org	09.03.2018 21:05	09.03.2018 21:05	15	15	✓
8	service.bund.de	09.03.2018 21:08	09.03.2018 21:09	30	30	✓
9	infsec.de	09.03.2018 21:12	09.03.2018 21:12	24	24	✓
10	cuckoosandbox.org	09.03.2018 21:15	09.03.2018 21:16	19	21	✓*

Tabelle 11.2: Manueller Vergleich zwischen WebChecker (A) und Google Chrome Browser (B). Alle Webseiten werden über https:// aufgerufen. *) Abweichungen, die durch den Ladevorgang des Favicons hervorgerufen werden.

11.2.3 *Analysegeschwindigkeit*

Überblick

Zur Messung der Geschwindigkeit wird eine Zeitmessung bei Abruf von zehn Webseiten durchgeführt. Verwendet wurde ein Server mit Intel(R) Core(TM) i7-5820K 6x3,3GHz CPU und 128 GB RAM. Dem Hostsystem, in dem Cuckoo und DisTrack ausgeführt werden, wurden 4 Kerne und 8 GB RAM zugewiesen. Dem Gastsystem stehen 2 Kerne und 2 GB RAM zur Verfügung.

Methodik

In der Messung werden die drei folgenden Phasen unterschieden: die Vor-

bereitung, die Browsermessung und die Abschlussphase. Die Vorbereitung umfasst die Initialisierung des Gastsystems und die Bereitstellung des gewünschten Snapshots der virtuellen Maschine. Sobald der Browser gestartet ist, beginnt die zweite Phase (Browsermessung), die bis zur vollständigen Beendigung des Browsers andauert. In der dritten Phase (Abschluss) werden die Messergebnisse verarbeitet und ein Ergebnisbericht erstellt.

Ergebnis

Es zeigt sich eine Abhängigkeit in der Dauer der Verarbeitungsphase von der Datenmenge (Tabelle 11.1). Durchschnittlich dauert ein vollständiger Durchlauf 4:06 Minuten. Die Erstellung von neuen Snapshots oder die Sicherung des Browserprofils wirkt verlängernd auf die Analysedauer ein. Die Messergebnisse können in Tabelle 11.3 eingesehen werden. Der langsame Seitenaufbau im Browser wird durch die Injection-Methoden begründet, wie sie in Abschnitt 9.2.4 beschrieben wurden. Durch Abschalten der Verhaltensüberwachung reduziert sich die Betrachtungszeit (Spalte Browser) auf wenige Sekunden.

Nr	Webseite	Startzeitpunkt	Vorbereitung	Browser	Abschluss	Gesamt
1	google.com	10.03.2018 15:23	36 Sek.	26 Sek.	72 Sek.	2:14 Min.
2	reddit.com	10.03.2018 15:28	38 Sek.	60 Sek.	152 Sek.	4:10 Min.
3	ebay.com	10.03.2018 15:36	35 Sek.	52 Sek.	140 Sek.	3:47 Min.
4	msn.com	10.03.2018 15:43	37 Sek.	30 Sek.	80 Sek.	2:27 Min.
5	washingtonpost.com	10.03.2018 15:51	35 Sek.	78 Sek.	189 Sek.	5:02 Min.
6	bbc.com	10.03.2018 16:01	36 Sek.	71 Sek.	194 Sek.	5:01 Min.
7	spiegel.de	10.03.2018 16:09	35 Sek.	84 Sek.	249 Sek.	6:08 Min.
8	yahoo.com	10.03.2018 16:18	35 Sek.	43 Sek.	108 Sek.	3:06 Min.
9	wordpress.com	10.03.2018 16:23	36 Sek.	36 Sek.	96 Sek.	2:48 Min.
10	nytimes.com	10.03.2018 16:29	36 Sek.	101 Sek.	249 Sek.	6:26 Min.

Tabelle 11.3: Messungen der Laufzeit. Alle Webseiten werden über https:// aufgerufen.

11.3 VERGLEICH MIT BESTEHENDEN WERKZEUGEN

Auswahl

In Abschnitt 9.5.1 wurden verschiedene bestehende Werkzeuge vorgestellt. Dabei zeigte sich OpenWPM als Werkzeug mit dem höchsten Reifegrad. Es wird in quantitativen Analysen am häufigsten eingesetzt. Aus diesem Grund findet im Folgenden ein qualitativer Vergleich der erhobenen Daten mit OpenWPM statt.

Abgrenzung

Wie in der Einleitung beschrieben besteht ist, keine direkte Konkurrenz zwischen DisTrack und OpenWPM, weil der Einsatz von OpenWPM in DisTrack möglich ist. Bleibt der Browser unverändert und werden nur browsereigene Funktionen zur Messung verwendet, zeigen sich jedoch Gemeinsamkeiten und Unterschiede in den Ansätzen. In diesem Abschnitt werden diese näher betrachtet.

Methodik

OpenWPM[3] wurde zu diesem Zweck auf einem Ubuntu 16.04.4-Desktop System mit Stand vom 20.03.2018 installiert. Als Testmenge dienen die Webseiten aus Abschnitt 11.2.2. Diese Webseiten werden mittels WebChecker und OpenWPM zeitgleich analysiert und die erhobenen Daten verglichen. Die Vergleichsmethodik orientiert sich an der Datenbankstruktur von OpenWPM und dem Ergebnisbericht von DisTrack (Abschnitt 9.4.4).

11.3.1 *Aufbau der OpenWPM-Datenablage*

Nach der Durchführung einer Analyse mit OpenWPM werden Messdaten in den folgenden Datenbank-Tabellen abgelegt:

- `http_requests`,
- `http_responses`,
- `http_redirects`,
- `localStorage`,
- `flash_cookies`,
- `profile_cookies`,
- `javascript_cookies` und
- `javascript`.

Datenbank

HTTP-basierte Anfragen und Antworten finden sich in den Tabellen `http_requests` und `http_responses`. Der Inhalt der Tabelle `http_redirects` wird den Feldern der Tabelle `http_responses` abgeleitet. Cookies und vergleichbare persistente Speicherformen sichert OpenWPM in den Tabellen `localStorage`, `flash_cookies`, `profile_cookies` und `javascript_cookies`. Die Tabelle `javascript` ermöglicht die Durchsicht von JavaScript-Methodenaufrufen, die während der Auswertung der Webseite getätigt werden.

Sonstige Daten

Weitere Tabellen dienen dem Management der Tests (Beschreibung, Testzeitpunkt, etc.), enthalten keine Messdaten und werden aus diesem Grund nicht weiter betrachtet. Neben der Speicherung in der Datenbank werden Screenshots des Webbrowsers sowie eine Log-Datei des Vorgangs im Dateisystem abgelegt.

11.3.2 *Vergleich der HTTP-Aufzeichnungen*

Vergleich

Zur Analyse von Web-Tracking sind die aufgezeichneten HTTP-Verbindungen von besonderer Bedeutung. Beide Werkzeuge führen eine vollständige Aufzeichnung der HTTP-Verbindungen durch. Einzige Ausnahme zeigt sich in der Aufzeichnung des Ladevorgangs das Favicons, wie bereits in Abschnitt 11.2.2 bemerkt wurde. In Tabelle 11.4 werden die Anzahl der HTTP-Anfragen gegenübergestellt.

Ergebnis

Die Tabelle `http_requests` in OpenWPM enthält zusätzliche Informa-

3 https://github.com/citp/OpenWPM/tree/e2e7dcdfc6b18dcbf73196fffd6a00d610d34a13, abgerufen am 20.03.2018.

Nr	Webseite	Zeitpunkt A	Zeitpunkt B	Anzahl A	Anzahl B	Bestanden
1	hochschule-trier.de	20.03.2018 19:35	20.03.2018 19:26	39	41	✓*
2	wikipedia.org	20.03.2018 19:38	20.03.2018 19:26	7	9	✓*
3	arbeitsagentur.de	20.03.2018 19:42	20.03.2018 19:26	28	30	✓*
4	blog.fefe.de	20.03.2018 19:55	20.03.2018 19:26	1	2	✓*
5	netzpolitik.org	20.03.2018 19:58	20.03.2018 19:26	60	62	✓*
6	kit.edu	20.03.2018 20:04	20.03.2018 19:26	56	59	✓*
7	www.kernel.org	20.03.2018 20:07	20.03.2018 19:26	14	16	✓*
8	service.bund.de	20.03.2018 20:11	20.03.2018 19:27	29	31	✓*
9	infsec.de	20.03.2018 20:15	20.03.2018 19:27	24	24	✓
10	cuckoosandbox.org	20.03.2018 20:19	20.03.2018 19:27	19	21	✓*

Tabelle 11.4: Manueller Vergleich zwischen WebChecker (A) und OpenWPM (B). Alle Webseiten werden über https:// aufgerufen. *) Abweichungen durch den Ladevorgang des Favicons begründet.

tionen zu auslösenden Ereignissen, beispielsweise welches JavaScript den HTTP-Request erzeugt hat. Diese internen Browserinformationen werden im HAR-Log nicht erfasst. Der Vergleich der aufgezeichneten HTTP-Header brachte keine Unterschiede hervor. Die Informationen der Tabelle `http_response` lassen sich vollständig aus dem HAR-Log extrahieren, mit Ausnahme des `is_cached`-Feldes. Bei Anfragen und Antworten werden im HAR-Log zusätzlich Zeitinformationen (`timings`) zur Verfügung gestellt. Diese Informationen sind in OpenWPM nicht verfügbar.

11.3.3 *Vergleich der Cookieinformationen*

Erfassungsmethodik

Die Erfassung der HTTP-Cookies erfolgt in DisTrack durch Öffnen und Verarbeiten der `cookies.sqlite`-Datei. OpenWPM setzt dies in gleicher Weise um und speichert die Werte in der Tabelle `profile_cookies`. Für die analysierten Webseiten wurde die gleiche Menge an Cookies gesetzt. Da in beiden Werkzeugen dieselbe Methodik angewendet wurde, war eine Übereinstimmung zu erwarten. Die Tabelle `javascript_cookies` ermöglicht die Einsicht, welche Cookies mittels JavaScript gesetzt wurden. DisTrack abstrahiert von der Art wie Cookies gesetzt werden. Sofern notwendig, kann diese Information durch einen Vergleich von Profilcookies und übermittelten HTTP-Headern rekonstruiert werden.

Localstore

So wie Cookies durch die Betrachtung der Datenbank ausgelesen werden, lässt sich der Localstore durch Öffnen der `webappsstore.sqlite`-Datenbank betrachten. Die getesteten Webseiten nehmen keine Speicherung im Localstore vor – dies konnte allerdings bei anderen Tests schon eingesetzt werden.

Adobe Flash

Grundsätzlich ermöglicht die DisTrack-Umgebung den Einsatz von Adobe Flash und eine Detektion von Schreibvorgängen in den entsprechenden Systemverzeichnissen. Dies wurde innerhalb der Voruntersuchung [189] de-

monstriert. In der hier vorgestellten Version von DisTrack wurde auf den Einsatz von Flash verzichtet, da:

- die Ausführung von Flash-basierten Inhalten vom Mozilla Firefox Browser unterbunden und erst durch eine Nutzeraktion gestartet wird,
- das Produktende vom Hersteller für 2020 angekündigt wurde und
- die Analyse von Flash innerhalb der Sandbox zu unerwarteten Systemabbrüchen geführt hat,
- es aufgrund von Sicherheitsbedrohungen und der abnehmenden technischen Unterstützung nur noch einen geringen Stellenwert einnimmt.

Bei einem Einsatz von OpenWPM innerhalb der DisTrack-Umgebung sollte aus den gleichen Gründen auf den Einsatz von Flash verzichtet werden.

11.3.4 *Weitere Gemeinsamkeiten und Unterschiede*

JavaScript

Die Tabelle `javascript` in OpenWPM ermöglicht eine Betrachtung von JavaScript-Methodenaufrufen. Diese können insbesondere zur Auswertung von aktivem Fingerprinting genutzt werden. DisTrack abstrahiert von solchen internen Abläufen. Eine Möglichkeit, diese Daten in zukünftigen Entwicklungen auch auf andere Art zu erfassen, wird in Abschnitt 11.8 aufgezeigt.

Netzwerk

OpenWPM bietet keine Aufzeichnung von Netzwerkdaten außerhalb des Browsers. Weitere Protokolldaten wie DNS oder OCSP werden nicht aufgezeichnet, worauf in der Dokumentation[4] von OpenWPM hingewiesen wird. In Quelltext 11.3.4 sind die Ergebnisse der Netzwerkmessung nach Abfrage der ersten Testseite `https://hochschule-trier.de` dargestellt. Es zeigen sich TCP-Anfragen an 6 weitere Hosts, die durch den Browser hervorgerufen werden und in OpenWPM unerkannt bleiben. So auch Anfrage und Inhalt von OCSP in Zeile 3. Vor Etablierung der Verbindung sind 13 Anfragen an den DNS-Server erforderlich. Diese UDP-Datagramme werden ebenfalls aufgezeichnet und sind im Ergebnisbericht bereitgestellt.

```
1 143.93.54.114 => {'hosts': ['hochschule-trier.de', 'www.
      hochschule-trier.de']}
2 34.208.65.55 => {'hosts': ['aus5.mozilla.org', 'balrog-aus5.r53
      -2.services.mozilla.com']}
3 93.184.220.29 => {'hosts': ['ocsp.digicert.com']}
4 52.85.245.159 => {'hosts': ['tracking-protection.cdn.mozilla.net'
      , 'd1zkz3k4cclnv6.cloudfront.net']}
5 52.34.107.172 => {'hosts': ['tiles.r53-2.services.mozilla.com', '
      tiles.services.mozilla.com']}
6 52.39.109.100 => {'hosts': ['shavar.services.mozilla.com', '
      shavar.prod.mozaws.net']}
7 52.85.245.16 => {'hosts': ['dcky6u1m8u6el.cloudfront.net', 'tiles
      -cloudfront.cdn.mozilla.net']}
```

4 https://github.com/citp/OpenWPM/tree/e2e7dcdfc6b18dcbf73196fffd6a00d610d34a13, abgerufen am 29.03.2018.

Netzwerkanfragen

Die Zeilen 3, 4, 5 und 6 der aufgeführten Anfragen sind Bestandteil des Browser Startprozesses. So beispielsweise die Prüfung, ob Programmaktualisierungen verfügbar sind (`aus5.mozilla.or`), oder den Abgleich von internen Listen wie beispielsweise die Safe-Browsing-Funktion (`shavar.services.mozilla.com`). Diese soll einen Schutz vor Phishing-Angriffen ermöglichen[5].

Sonstige Anfragequellen

Grundsätzlich kann auch das Betriebssystem als Auslöser für Netzwerk-Anfragen in Frage kommen. Durch die Nutzung eines Wiederherstellungspunktes der virtuellen Maschine sind diese allerdings reproduzierbar und in den bisherigen Tests nicht aufgetreten. Durch Deaktivierung der automatischen Update-Funktionen wurde das Risiko von Systemanfragen in Analysevorgängen minimiert.

11.4 Analysemodelle und Ergebnisse

Alleinstellungsmerkmale

In diesem Abschnitt werden Analysemodelle vorgestellt, welche die besonderen Vorteile der Sandbox-basierten Analyseumgebung ausnutzen. Ohne umfangreiche Erweiterungen und Anpassungen lassen sich diese Tests nicht mit den vorgestellten Werkzeugen aus verwandten Arbeiten durchführen. Somit stellen sie die Alleinstellungsmerkmale des DisTrack-Frameworks und Vorteile der Sandboxumgebung zur Datenerfassung dar. Die vorgestellten Ergebnisse werden ohne Rückgriff auf die Ergebnisse des WebCheckers bzw. der HAR-Log basierten Analyse erbracht. Sie lassen sich auf jedes andere Messinstrument und Browser übertragen.

11.4.1 *Modell I: DNS-Analyse*

DNS

Namensauflösungen von Hostadressen werden durch das DNS-Protokoll realisiert. Sofern die Netzwerkadresse eines Hosts unbekannt ist, findet eine Abfrage des DNS-Servers statt. Im Erfolgsfall liefert dieser eine oder mehrere IP-Adressen zurück. Diese Auflösung wird für einen gewissen Zeitraum vom System zwischengespeichert. Bei weiteren Anfragen an den bereits aufgelösten Host wird keine erneute Auflösung benötigt. Auf diese Weise können kürzere Zugriffszeiten realisiert werden.

Passives FP

In den Abschnitten 7.3.3 und 8.2.5 wurde die Möglichkeit von DNS-basiertem Tracking beschrieben. Von Daniel Dent wird diese Thematik durch einen Demonstrator verdeutlicht, der unter http://dnscookie.com/ abgerufen werden kann. Ein Screenshot der Webseite ist in Abbildung A.3 in Anhang A zu sehen. Durch zufällige Zuweisungen von IP-Adressen zu einem Host setzen sich die Bits eines 32-Bit langen Cookies zusammen. Bei erneutem Besuch der Webseite wird die lokale Pufferung genutzt, wodurch das gleiche Cookie erzeugt wird.

Pufferung

Eine solche Pufferung stellt eine Markierung des Systems dar, die für Web-

5 https://support.mozilla.org/de/kb/wie-funktioniert-schutz-vor-betrugsversuchen-und-schadprogrammen, abgerufen am 26.03.2018.

Tracking ausgenutzt werden kann. Anders als übliche Tracking-Verfahren wirkt sie nicht auf den Browser, sondern auf das umliegende System ein. Aus diesem Grund kommen Systemzustände zum Einsatz.

Modell

Modellaufbau

Im ersten Schritt wird ein neues Browserprofil mit Abruf einer leeren Webseite (about:blank) erzeugt. Dies kommt in allen folgenden Schritten zum Einsatz. Im zweiten Schritt wird die Webseite auf Basis des neu erzeugten Browserprofils abgerufen. Nach Beendigung des Browsers und Sicherung aller Daten wird der Systemzustand durch einen Snapshot festgehalten, der im dritten Schritt als Startpunkt dient. In diesem wird die Webseite erneut abgerufen und die Anzeige des Cookies abgewartet. Der vierte Schritt ist von den Schritten 2 und 3 entkoppelt und verwendet das Browserprofil aus dem ersten Schritt. Wie in Schritt 2 wird die Webseite abgerufen und ein Snapshot erzeugt. Dieser Snapshot wird im anschließenden Schritt 5 genutzt.

Ziel

Es ist zu erwarten, dass das generierte DNS-Cookie in Schritt 2 dem in Schritt 3 entspricht, sich aber von dem in den Schritten 4 und 5 unterscheidet. Durch die Nutzung des gleichen Browserprofils (aus Schritt 1) kann gezeigt werden, dass dieser nicht als Informationsträger dient, sondern allein für das System verantwortlich ist.

```
1  def analysis(self, analysis):
2  t1 = analysis.add_step()
3  p1 = t1.get_profile_after()
4
5  t2 = analysis.add_step(["http://dnscookie.com/"], wtdprofile=p1)
6  s2 = t2.get_state_after()
7
8  t3 = analysis.add_step(["http://dnscookie.com/"], wtdprofile=p1,
       wtdstate=s2)
9
10 t4 = analysis.add_step(["http://dnscookie.com/"], wtdprofile=p1)
11 s4 = t4.get_state_after()
12
13 t5 = analysis.add_step(["http://dnscookie.com/"], wtdprofile=p1,
       wtdstate=s4)
14
15 analysis.start()
```

Quelltext 11.1: Umsetzung des Modells aus Abbildung 11.1 in DisTrack.

Modellumsetzung

Das Modell wird in Abbildung 11.1 dargestellt und durch Quelltext 11.1 kodiert. In Zeile 2 und 3 wird das Ausgangsprofil für Schritt 1 erzeugt. In Zeile 5 wird die Webseite erstmalig abgerufen und der Systemzustand durch Zeile 6 gesichert. In Zeile 8 wird das Ausgangsprofil aus Schritt 1 in Verbindung mit dem Systemzustand nach Ausführung von Schritt 2 verwendet. Die Schritte 4 und 5 sind analog zu den Schritten 1 und 2 aufgebaut, was sich durch die ähnlichen Befehle der Zeilen 9 bis 13 widerspiegelt.

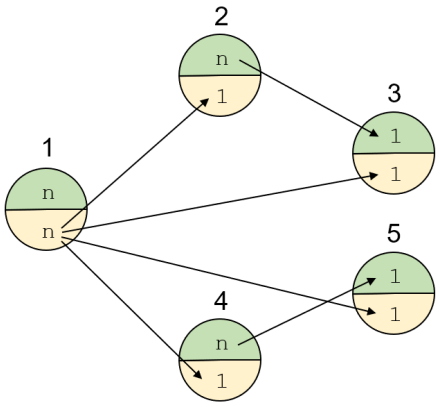

Abbildung 11.1: Modell I: Übersicht und Ablauf. Legende wie Abbildung 10.2.

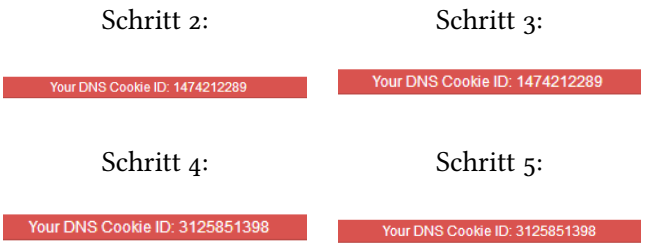

Abbildung 11.2: Screenshot von dnscookie.com nach Ausführung der Schritte 2-5.

Auswertung und Ergebnis

Auswertung

Eine Auswertungsform stellt die Betrachtung des Cookie-Wertes der Webseite dar. Durch den gespeicherten Screenshot des WebCheckers (Abschnitt 10.6.1) kann dieser betrachtet werden. Die Ergebnisse sind in Abbildung 11.2 dargestellt. Dabei zeigt sich die erwartete Übereinstimmung der Schritte 2 und 3 sowie 4 und 5.

Netzwerkdaten

Darüber hinaus sollen durch eine Datenanalyse die Unterschiede sichtbar werden. Aus diesem Grund werden die aufgelösten DNS-Adressen betrachtet, die durch die Netzwerkanalyse erfasst werden. In Quelltext 11.2 wird ein solche Auswertung gezeigt. In den Zeilen 5 bis 8 werden die Ergebnisse des Netzwerkmittschnitts ausgelesen. In den Zeilen 5 und 8 ist zu sehen, dass die Ergebnisse von Schritt 2 und Schritt 4 verglichen werden. Ab Zeile 10 startet die Vorbereitung der Ausgabe von beiden Messergebnissen. So werden zunächst die DNS-Abfragen in Zeile 11 sortiert. Zeile 11 iteriert über alle nun sortierten DNS-Abfragen und Zeile 12 filtert nach Namensauflösungen zu IP-Adressen. Die Zeilen 14 und 15 extrahieren den Host und die erste Antwort des DNS-Servers. Grundsätzlich müssen hier alle Antworten bewertet werden, allerdings liefert in diesem Fall der Server nur eine Antwort, weshalb nur diese ausgewertet wird. Zeile 16 dient der Ausgabe von Anfrage und Antwort.

Ergebnisse

Die relevanten Ergebnisse der Ausgabe sind in Tabelle 11.5 abgebildet. Wie zu erkennen ist, unterscheiden sich die Antworten des DNS-Servers im letzten Oktett der IP-Adresse.

```
def report(self, analysis):

        t1,t2,t3,t4,t5 = analysis.get_steps()

        tr2 = self.get_task_report(t2)
        tdns2 = tr2.get_network_report()['dns']
        tr4 = self.get_task_report(t4)
        tdns4 = tr2.get_network_report()['dns']

        for t in [tdns2,tdns4]:
            sortreq = sorted(t, key=lambda k: k['request'])
            for dnsreq in sortreq:
                if dnsreq['type'] == 'A':
                    request = dnsreq['request']
                    answer = dnsreq['answers'][0]['data']
                    print "=>".join([request,answer])
```

Quelltext 11.2: Auswertung der Messergebnisse nach Ausführung von Quelltext 11.1.

Nutzen

Weil nicht exakt bestimmt werden kann, wie lange ein solches Cookie aktiv bleibt, wird diese Form des Trackings nicht der Stabilitätsanforderung aus Abschnitt 7.1.3 gerecht. Je nach DNS-Server, der die rekursive Namensauflösung durchführt, könnten auf verschiedenen Systemen das gleiche Cookie erzeugt werden. Dies würde der Forderung nach Eindeutigkeit widersprechen.

Fazit

Auch wenn sich eine solches Verfahren als langfristiger Cookie-Ersatz eher ausschließt, muss berücksichtigt werden, dass eine Stabilität über eine kurze Zeitspanne bereits kritisch bewertet werden muss. Nach manueller Löschung von Cookies, Cache und Historie werden diese Merkmale durch Cookie-Respawning (nach Abschnitt 7.3.4) wiederhergestellt.

11.4.2 *Modell II: Manuelle Kontextlöschung*

Motivation

In Kapitel 7 wurde in Abschnitt 7.7 das Kontextmanagement als Schutz vor Web-Tracking beschrieben. Durch eine Löschung von Cookies, Cache und Browserverlauf soll die Identifikation des Nutzers und die damit verbundene Profilbildung durch den Tracker verhindert werden.

Fragestellung

Es fragt sich, ob der Browser eine solche Bereinigung vollständig durchführt. Sofern der vom Browser genutzte Speicher vollständig gelöscht wird, wäre eine Identifikation auf Basis von speicherbasierter Verfahren nicht länger möglich. Bleiben allerdings Datenreste bestehen, können diese möglicherweise für Web-Tracking-Verfahren ausgenutzt werden.

Ansatz

Eine Prüfung der gelesenen und geschrieben Dateien auf dem Datenträger kann mittels DisTrack durchgeführt werden. So lässt sich beobachten, auf welche Daten während eines Webseitenbesuchs zugegriffen und ob diese bei einer manuellen Bereinigung gelöscht werden. Die manuelle Löschung wird

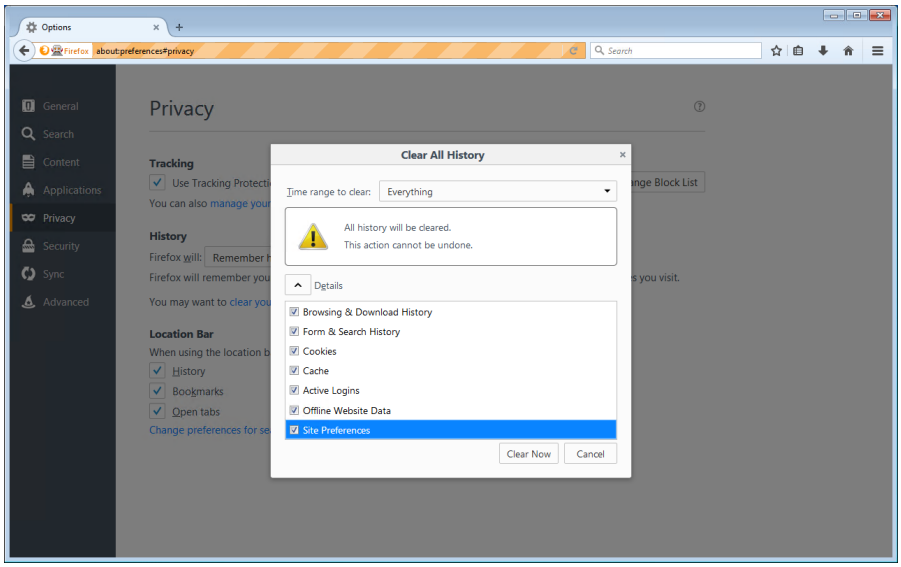

Abbildung 11.3: Löschung der Historie im Mozilla Firefox Browser.

über das Nutzerinterface des Browsers ausgeführt, wie in Abbildung 11.3 zu sehen ist.

Randbedingungen

Bei dieser Form des Tests besteht eine Abhängigkeit von der Testmenge. Sofern eine Webseite kein Web-Tracking durchführt, werden möglicherweise bestimmte Speicherformen nicht genutzt und treten deshalb nicht in Erscheinung. Diese Abhängigkeit lässt sich durch den Aufruf mehrerer Webseiten reduzieren. Es ist wichtig einzusehen, dass auf diese Weise kein Beweis erbracht wird, dass die Löschung in jedem Fall wirksam ist. Allerdings lassen sich auf diesem Weg Beweise dafür finden, unter welchen Umständen kein Effekt erzielt wird.

Modell

Modellaufbau

Zunächst wird ein neues, aktuelles und ungenutztes Browserprofil benötigt. Ein solches wird durch Öffnung einer leeren Webseite (about:blank) erstellt. Das neue Browserprofil wird gesichert und dient als Ausgangspunkt des nächsten Schrittes. In Schritt 2 wird die zu testende Webseite aufgerufen. Wie bereits beschrieben, besteht ein Abhängigkeitsverhältnis zur gewählten Webseite. Zu Demonstrationszwecken wird an dieser Stelle die Webseite der Washington Post[6] geöffnet. Im dritten Schritt wird der Browser gestartet und das Browserprofil nach dem Webseitenbesuch aus Schritt 3 verwendet. Nun findet eine Unterbrechung der Ausführung statt, um eine manuelle Löschung der Kontextdaten durchzuführen. Im Browser wird das in Abbildung 11.4 gezeigte Menü aufgerufen und ausgeführt. Wichtig ist, dass hierbei alle Datenarten ausgewählt werden, da dies nicht der Voreinstellung entspricht. Nach Beendigung des Browsers wird erneut das Browserprofil gesichert. Im vierten und letzten Schritt findet ein erneuter Besuch der

6 https://washingtonpost.com, abgerufen am 12.03.2018.

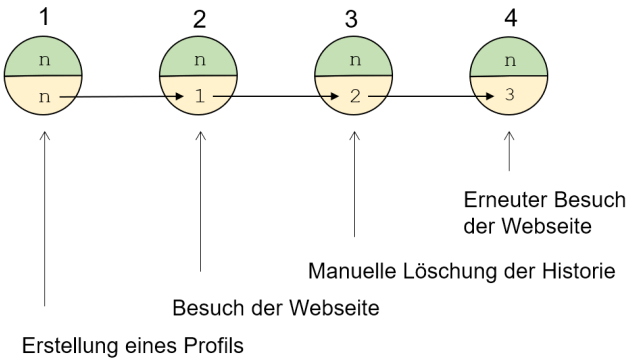

Abbildung 11.4: Modell II: Übersicht und Ablauf. Legende wie Abbildung 10.2.

Webseite statt. Es werden erneut alle geschriebenen und gelesenen Dateien gemessen. Somit ist der Zugriff auf Daten von Interesse, die nach Löschung im vorherigen Schritt noch bestehen geblieben sind.

Modellumsetzung

Das hier beschriebene Modell ist in Abbildung 11.4 entsprechend der Notation aus Abschnitt 10.4.2 visualisiert. Quelltext 11.3 zeigt die Umsetzung des Modells in DisTrack. Es besteht aus den vier beschriebenen Schritten. Eine Besonderheit stellt die Zeile 11 dar, in der mittels `set_waitafter()` ein Haltepunkt gesetzt wird. Die Ausführung von DisTrack hält an dieser Stelle an und ermöglicht die manuelle Interaktion mit dem Browser. Nach einer Bestätigung durch den Nutzer (von DisTrack) wird die Ausführung fortgeführt. In Zeile 14 wird DisTrack angewiesen, den Profilzustand zu sichern, da dieser ansonsten verworfen werden würde.

Ausführung

Der Test wurde am 11.03.2018 um 13:40 Uhr gestartet und die Messung dauerte 20 Minuten. Während der Ausführung sind keine unerwarteten Fehler aufgetreten.

```
1  def analysis(self, analysis):
2
3          t1 = analysis.add_step()
4          p1=t1.get_profile_after()
5
6          t2 = analysis.add_step(["https://washingtonpost.com"],
               wtdprofile=p1)
7          p2=t2.get_profile_after()
8
9          t3 = analysis.add_step(wtdprofile=p2)
10         p3=t3.get_profile_after()
11         t3.set_waitafter()
12
13         t4 = analysis.add_step(["https://washingtonpost.com"],
               wtdprofile=p3)
14         t4.keep_profile()
15
16         analysis.start()
```

Quelltext 11.3: Umsetzung des Modells aus Abbildung 11.4 in DisTrack.

Ziele

Für die Auswertung der Daten wird die Verhaltensanalyse von Schritt 2 und Schritt 4 betrachtet. Im Fokus stehen die neu erstellten und gelesenen Dateien. Kernidee der Auswertung ist die Prüfung, welche Dateien in Schritt 2 erstellt werden, in Schritt 3 nicht gelöscht und in Schritt 4 gelesen werden.

Auswertung

Die Auswertung wird in Quelltext 11.4 skizziert und kann im Anhang in Quelltext C.5 vollständig eingesehen werden. Die Zeilen 1-9 erheben einen Ergebnisbericht von der Sandbox. In den Variablen `summary1` und `summary2` befinden sich Auflistungen der erstellten, geschriebenen und gelesenen Dateien aus Schritt 2 und Schritt 4. Zeile 11 iteriert über alle gelesenen Dateien in Schritt 4, die in Schritt 2 und nicht in Schritt 4 erstellt wurden (Zeile 12). Sofern in Schritt 4 noch Dateien aus Schritt 2 bestehen die in Schritt 2, fand keine Löschung in Schritt 3 statt. Zusätzlich wird in den Zeilen 13 und 14 noch ein Hash erstellt, um einen inhaltlichen Vergleich durchzuführen.

Umsetzung der Löschung

Ein offenes Problem ist, dass die Löschung im dritten Schritt auch durch eine inhaltliche Änderung durchgeführt werden kann. So wäre es möglich, dass die Dateien nicht gelöscht, aber beispielsweise mit nullen oder einem zufälligen Inhalt überschrieben wurden, was eine Äquivalenz zur Löschung darstellt. Deshalb wird in Zeile 15 ein Delta der Datei aus Schritt 3 und Schritt 4 gebildet. Umgesetzt wird dieser Vergleich durch die Programmbibliothek fuzzywuzzy[7], welche den Grad der Übereinstimmung einem Wert zwischen 0 (ungleich) und 100 (identisch) zuordnet. Berechnet wird dies auf Basis der Levenshtein-Distanz [104].

```
1  def report(self, analysis):
2
3          t1,t2,t3,t4 = analysis.get_steps()
4
5          a = self.get_task_report(t2)
6          summary1 = a.behavior.get_firefox_summary().short_summary
7
8          a = self.get_task_report(t4)
9          summary2 = a.behavior.get_firefox_summary().short_summary
10
11         for rw in summary2['file_read']:
12             if rw in summary1['file_created'] and rw not in
                   summary2['file_created']:
13                 h1=t2.get_profile_after().get_hash_file(rw)
14                 h2=t3.get_profile_after().get_hash_file(rw)
15                 dist= str(fuzz.ratio(t2.get_profile_after().
                       get_profile_file(rw), t3.get_profile_after().
                       get_profile_file(rw)))
16                 self._output(rw,h1,h2,dist)
```

Quelltext 11.4: Auswertung der Messergebnisse nach Ausführung von Quelltext 11.3.

Ergebnis

Nach der Ausführung von Quelltext 11.4 können die gemeldeten Dateien näher betrachtet werden. Die vollständige Ausgabe ist im Anhang in Quell-

7 https://pypi.python.org/pypi/fuzzywuzzy, abgerufen am 12.03.2018

text C.10 zu finden. An dieser Stelle wird nur das bedeutendste Finding in Quelltext 11.5 näher betrachtet[8].

Verbleibende Dateien

Es zeigen sich zwei Dateien, die während des Besuchs in Schritt 2 geschrieben werden, aber nach Löschung in Schritt 3 weiterhin bestehen und in Schritt 4 zugegriffen werden. Somit ist klar, dass diese Speicherform für Web-Tracking genutzt werden kann und diese eine manuelle Löschung innerhalb des Browsers überdauert.

```
1 ################################################################
2 [TP]\storage\default\https+++www.washingtonpost.com\idb
      \12183338011.sqlite
3 Hash gleich
4 Levenshtein-Distanz: 100
5 ################################################################
6 [TP]\storage\default\https+++www.washingtonpost.com\idb\301792106
      ttes.sqlite
7 Hash gleich
8 Levenshtein-Distanz: 100
```

Quelltext 11.5: Ausschnitt der Ergebnisse nach Ausführung von Quelltext 11.4.

Schwachstelle

Bereits der Name des Ordners offenbart, welche Webseite für die Speicherung verantwortlich ist. Der Ordner `idb` zeigt, dass es sich um die IndexedDB-Funktion handelt. Dabei stellt der Browser der geöffneten Webapplikation eine Key-basierte Datenbank[9] zur Verfügung, die im Firefox mit einer *sqlite*-Datenbank persistent gespeichert wird. Der Test hat hervorgebracht, dass diese Speicherform durch einen manuellen Bereinigungsvorgang nicht berücksichtigt wird.

Fazit

Dieser Umstand ist seit längerer Zeit bekannt[10], wurde allerdings bis zur hier genutzten Browserversion (55.0.2) nicht behoben. Durch DisTrack konnte dies erstmalig systematisch nachgewiesen werden.

11.4.3 *Modell III: Font-basiertes Fingerprinting*

Motivation

In Abschnitt 7.4 wurde beschrieben, wie durch aktive Verfahren ein Fingerabdruck vom Browser erzeugt ist. Je individueller ein System konfiguriert wird, desto stärker unterscheidet sich die Konfiguration von anderen Nutzern und ermöglicht so eine eindeutige Wiedererkennung. Neben diversen Informationsquellen werden auch die installierten Schriftarten zur Erzeugung des Fingerabdrucks ausgelesen. Während die Standardschriftarten wenig Unterscheidungsmerkmale liefern, führt die Installation von seltenen Schrifttypen zu Eindeutigkeit und Stabilität[11].

Demonstrator

Zur Generierung des Fingerabdrucks müssen die installierten Schriften

8 Zu beachten ist, dass der Pfad zum Browserprofil mittels [TP] (temporary profile) abgekürzt wird.

9 https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API, abgerufen am 12.03.2018.

10 https://bugzilla.mozilla.org/show_bug.cgi?id=527667, abgerufen am 12.03.2018.

11 Eindeutig und stabil im Sinne der Anforderungen aus Abschnitt 7.1.3.

vom Webseitenbetreiber ausgelesen werden. Zu diesem Zweck existieren verschiedene Methoden. Die Webseite https://browserleaks.com/fonts demonstriert dies unter Einsatz von JavaScript. In dieser Implementierung wird der Browser zur Nutzung von 512 Schriftarten angewiesen, wobei ein Skript prüft, in welchen Fällen dies erfolgreich war.

Grundidee

Damit der Browser die angeforderte Schrift darstellen kann, muss diese aus dem System geladen werden. Auf einem üblichen Windowssystem sind die Schriftarten unter `C:\Windows\Fonts` abgelegt. Infolgedessen wird erwartet, dass der Browser auf diese zugreifen muss, wenn die Schrift zur Darstellung der Webseite benötigt wird. Ein solcher Lesezugriff kann durch die Sandbox detektiert werden.

Modell

Ziel

Zur Überprüfung dieser Theorie wird ein weiteres DisTrack-Modell erstellt. Ziel ist der Vergleich eines normalen Webauftritts mit einer Webseite, auf der aktives Fingerprinting durchgeführt wird.

Modellaufbau

Im ersten Schritt muss ein neues Browserprofil erzeugt werden, welches als Ausgangsbasis für die weiteren Analyseschritte dient. Im zweiten Schritt wird eine Webseite aufgerufen, von der kein Web-Tracking erwartet wird. Diese Analyse soll als Baseline zum Vergleich dienen, weshalb es besonders wichtig ist, dass auf dieser kein aktives Fingerprinting stattfindet. In diesem Fall wurde sich für die Webseite https://wikipedia.org entschieden. Im dritten Schritt wird eine Webseite besucht, die ein solches Fingerprinting durchführt. Als Repräsentant für diese Webseiten wird https://browserleaks.com/fonts aufgerufen.

Modellumsetzung

Das hier beschriebene Modell wird in Abbildung 11.5 visualisiert. Die konkrete Implementierung der Analyse wird in Quelltext 11.6 veranschaulicht. Der Test wurde am 16.03.2018 um 19:55 Uhr gestartet und die Messung nahm 6 Minuten in Anspruch. Während der Ausführung sind keine unerwarteten Fehler aufgetreten.

```
1  def analysis(self, analysis):
2          t1 = analysis.add_step()
3          p1 = t1.get_profile_after()
4
5          t2 = analysis.add_step(["http://wikipedia.org"],
               wtdprofile=p1)
6          t3 = analysis.add_step(["http://browserleaks.com/fonts"],
                wtdprofile=p1)
7
8          analysis.start()
```

Quelltext 11.6: Umsetzung des Modells aus Abbildung 11.5 in DisTrack.

Auswertung und Ergebnis

Vergleich

Zur Auswertung der Messung werden die geöffneten Dateien aus Schritt 2

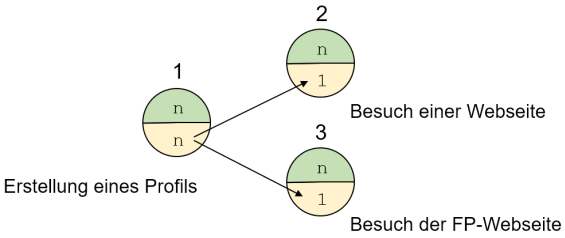

Abbildung 11.5: Modell III: Übersicht und Ablauf. Legende wie Abbildung 10.2.

und Schritt 3 verglichen. Interessant sind insbesondere die Dateien, die nur in Schritt 3 geöffnet wurden.

```
1  def report(self, analysis):
2
3          t1,t2,t3 = analysis.get_steps()
4
5          tr2 = self.get_task_report(t2)
6          sum2 = tr2.behavior.get_firefox_summary().short_summary
7
8          tr3 = self.get_task_report(t3)
9          sum3 = tr3.behavior.get_firefox_summary().short_summary
10
11         key='file_opened'
12         for a in sum3[key]:
13             if f not in sum2[key] and "Fonts" in f:
14                 print f
```

Quelltext 11.7: Auswertung der Messergebnisse nach Ausführung von Quelltext 11.6.

Auswertung

Quelltext 11.7 zeigt, wie zunächst die Verhaltensinformationen des Browsers von Schritt 2 und 3 in den Zeilen 5 bis 9 ausgelesen werden. Da nur die gelesenen Dateien betrachtet werden sollen, wird dies in Zeile 11 festgelegt. Die Zeilen 12 bis 14 ermöglichen die beschriebene Selektion von disjunkten Dateiöffnungen aus Schritt 3 des Modells. Um irrelevante Informationen auszublenden, werden nur die Dateien (f) angezeigt, die `Fonts` im Dateipfad aufweisen.

Ergebnis

Das Ergebnis nach Ausführung von Quelltext 11.7 ist eine Aufzählung von insgesamt 76 Schriftarten, die ausschnittsweise in Quelltext 11.8 abgebildet sind. Alle in Schritt 3 geladenen Schriften umfassen 114 Dateien. Sowohl im ersten als auch im zweiten Schritt werden 39 Schriftarten geladen. Auf dem Gastsystem sind 134 Schriftarten installiert.

```
...
C:\Windows\Fonts\msmincho.ttc
C:\Windows\Fonts\simfang.ttf
C:\Windows\Fonts\constan.ttf
C:\Windows\Fonts\upcll.ttf
C:\Windows\Fonts\simsunb.ttf
C:\Windows\Fonts\pala.ttf
C:\Windows\Fonts\upckl.ttf
C:\Windows\Fonts\daunpenh.ttf
C:\Windows\Fonts\david.ttf
C:\Windows\Fonts\Vani.ttf
C:\Windows\Fonts\rod.ttf
C:\Windows\Fonts\msuighur.ttf
C:\Windows\Fonts\taile.ttf
C:\Windows\Fonts\angsau.ttf
C:\Windows\Fonts\arabtype.ttf
C:\Windows\Fonts\cordia.ttf
C:\Windows\Fonts\ntailu.ttf
C:\Windows\Fonts\Candara.ttf
C:\Windows\Fonts\simpfxo.ttf
...
```

Quelltext 11.8: Ausschnitt der Ergebnisse nach Ausführung von Quelltext 11.7.

Fazit

Die Analyse zeigt, dass durch Prozessüberwachung die Durchführung von Font-basiertem Fingerprinting detektiert werden kann. Die Erfassung lässt sich unabhängig davon durchführen, mit welcher Technik (CSS, Javascript) die Prüfung stattfindet. Sofern eine Schriftart zur Darstellung einer Webseite genutzt wird, muss diese vom Browser geladen werden. Indikator für Font-basiertes Fingerprinting ist eine unverhältnismäßig umfangreiche Einbindung von Schrifttypen.

11.5 Weitere Modelle und Einsatzmöglichkeiten

Weiteres

Neben den hier vorgestellten Analysemodellen kommen weitere Analysen in Betracht, für die das DisTrack-Framework eingesetzt werden kann:

HSTS. Wie in Abschnitt 7.3.3 beschrieben und im Internet gemessen[12] wurde, können HSTS-Zertifikatsinformationen als Cookie eingesetzt werden. Für eine Detektion in DisTrack werden die Lese- und Schreiboperationen der `SiteSecurityServiceState.txt`-Datei im Profilordner des Mozilla Firefox überwacht.

ADDONS. Es ist unklar, welche Netzwerkaktivitäten von der Installation und Ausführung von Browsererweiterungen ausgehen. In Abschnitt 4.3.1 wurde beschrieben, wie in der Vergangenheit Browser-Toolbars zur Erhebung von Nutzungsanalysen eingesetzt wurden. Auch heute übermitteln Erweiterungen Informationen zur Nutzung an den Softwarehersteller[13].

12 https://www.heise.de/security/meldung/Verschluesselung-Apple-entdeckt-und-entschaerft-HSTS-Supercookies-3998754.html, abgerufen am 26.03.2018.

13 https://www.technologyreview.com/s/516156/a-popular-ad-blocker-also-helps-the-ad-industry/, abgerufen am 27.03.2018.

Forensik

Neben den hier vorgestellten Einsatzzwecken zur Analyse von Web-Tracking, kommt die vorstellte Technik zur Entwicklung forensischer Analysen in Frage. In Modell II in Abschnitt 11.4.2 wurde geprüft, welche Datenreste auf dem System nach einer manuellen Löschung des Browserkontextes verbleiben. Diese können in Quelltext C.10 eingesehen werden. Dabei zeigt sich, dass die Datei `cookies.sqlite-wal` sich nach einer Löschung der Cookies im Profilordner befindet. Bei `*-wal`-Dateien handelt es sich um eine „Write-Ahead Log"-Datei[14], die vom Datenbanksystem (sqlite) aus Gründen der Effizienz und Transaktionssicherheit erzeugt wird. Eine inhaltliche Betrachtung der Dateien offenbart Datenreste des ursprünglichen Datenbankinhalts. Es ist unwahrscheinlich, dass sich diese zum Web-Tracking nutzen lassen. Bei einer forensischen Analyse können sie allerdings Hinweise auf die gelöschten Daten liefern und manuell ausgewertet werden.

11.6 Invasivität des Browsers

Internet Explorer

Wie in Abschnitt 9.5 beschrieben, wurde in der publizierten Voruntersuchung [189] der Internet Explorer 8 in einer Sandbox-Umgebung während des Abrufs von populären Webseiten überwacht. Dabei sind umfangreiche Änderungen an der Windows-Registry aufgefallen. Dies ist ausschnittsweise in Abbildung A.1 in Anhang A zu sehen. Fraglich ist, welche lesenden und schreibenden Operationen vom Mozilla Firefox bei einem Webseitenabruf ausgehen.

Vergleich

Für diese Prüfung wurde die Testergebnisse aus Abschnitt 11.2.1 erneut betrachtet:

REGISTRY Es wurden auf zwei Registry-Schlüssel schreibend zugegriffen. Lesender Zugriff fand auf 930 Schlüssel statt.

DATEISYSTEM Es wurde auf 2270 Dateien schreibend zugegriffen. Dabei fand kein schreibender Zugriff auf Dateien außerhalb des Profilordners statt. Lesend wurde auf Systembibliotheken, Schriftarten und Programmdateien zugegriffen. Insgesamt wurden 69 DLL-Dateien geladen und ausgeführt.

NETZWERK Mit 418 IP-Adressen wurde eine Verbindung aufgebaut und 706 Hostnamen wurden über einen DNS-Server aufgelöst.

Ergebnis

Ein Vergleich mit der Voruntersuchung zeigt, dass der Mozilla Firefox weniger invasiv in das umliegende System eingreift als der Internet Explorer 8.

11.7 Prüfung der Anforderungen

Anforderungen

In Folgenden werden die Anforderungen aus Abschnitt 10.3 einer Prüfung auf Umsetzung in DisTrack unterzogen.

14 https://www.sqlite.org/wal.html, abgerufen am 27.03.2018.

(B-1) **Stabilität**. Die Prüfungen in Abschnitt 11.2.1 zeigten keine Stabilitätsprobleme[15]. Grundsätzlich kann das Werkzeug zur Durchführung von quantitativen Studien unter Berücksichtigung der gemessenen Laufzeiten eingesetzt werden.

(B-2) **Vollständigkeit**. Die von DisTrack erhobenen Daten werden über einen üblichen Browser erhoben. Mittels der Vergleichsprüfung in Abschnitt 11.2.2 konnte die Korrektheit der Messdaten sichergestellt werden. Durch die zusätzlichen Messinstrumente zur Netzwerk- und Verhaltensanalyse stehen noch weitere Daten für eine Auswertung zur Verfügung. Die Umgebung zeichnet hingegen keine browserinternen Abläufe auf. Aufrufe von JavaScript-Funktionen können auf diese Weise nicht erfasst werden.

(B-3) **Effizienz**. Nachteilig an dieser Analyseform sind die langen Laufzeiten, wie sie in Abschnitt 11.2.3 dargestellt sind. Initialisierung, Messung, Verarbeitung und Berichterstattung nehmen durchschnittlich vier Minuten in Anspruch. Eine gleichzeitige Ausführung mehrerer Analysen wird durch die Cuckoo Sandbox unterstützt und kann für Folgearbeiten in Aussicht gestellt werden.

(B-4) **Allgemeinheit**. Das Werkzeug lässt sich vielseitig für verschiedene Formen von Analysemodellen einsetzen, wie sie in Abschnitt 11.4 präsentiert wurden. Es ermöglicht umfassende Analysen und Auswertungen, die über bestehende Werkzeuge hinausgehen, wie in Abschnitt 11.3 gezeigt wurde. Die Möglichkeit, eigene Modelle und entsprechende Analysen zu entwickeln, ist neuartig und stellt eine Verbesserung zu bestehenden Werkzeugen dar.

(B-5) **Authentizität**. Der in Abschnitt 10.5.1 beschriebene WebChecker ermöglicht eine Analyse durch den Browser ohne Rückgriff auf Modifikationen oder Erweiterungen desselbigen. Dies ist bei einer verhaltensbasierten Analyse von besonderer Wichtigkeit und konnte durch browsereigene Messmittel ermöglicht werden.

15 Der Einsatz von Adobe Flash führte zu unerwarteten Systemabbrüchen und wurde aus diesem Grund nicht eingesetzt.

Forschungsfrage

Abschließend soll die Forschungsfrage aus Abschnitt 10.2 beantwortet werden:

> **Welchen Mehrwert bietet die verhaltensbasierte Analyse des Browsers zur Erkennung von Web-Tracking-Methoden?**

DisTrack

Der Einsatz einer Sandbox ermöglicht die Erfassung der Interaktion von Browser und umliegendem System. Die vorgestellten Modelle zeigen, dass Zustandsänderungen des umliegenden Systems für Web-Tracking genutzt werden können. Durch die Beobachtung von Speichervorgängen konnte systemisch die Wirkungslosigkeit einer manuellen Löschung des Browserkontexts nachgewiesen werden. Auch der Zugriff auf bestimmte Systemressourcen kann als Indikator für spezielle Tracking-Verfahren dienen. Die vollständige Netzwerkkommunikation und die Beobachtung von Systemveränderungen entziehen sich der Erfassung durch bisherige Werkzeuge und stellen eine Stärke dieser Analyseform dar. Im Ausblick stehen weitere Möglichkeiten des Einsatzes, wie eine Überprüfung der Aktivitäten von Browsererweiterungen.

WebChecker

Der WebChecker zeigt im Vergleich mit verwandten Messwerkzeugen (Abschnitt 11.3) nur geringe Abweichungen in Qualität und Quantität der HTTP-bezogenen Messdaten: Dies umfasst die Erfassung der Kommunikation über HTTP und die Auswertung von gesetzten Cookies. Basierend auf diesen Daten lassen sich beliebige Auswertungen durchführen und stellen somit ein allgemeines Messinstrument gemäß Anforderung (B-4) dar. Sofern interne Browserabläufe erhoben werden sollen, ist dies mit browsereigenen Mitteln nicht umsetzbar. In diesem Fall ist der Einsatz von OpenWPM anstelle des WebCheckers möglich.

Nachteile

Unabhängig von der Wahl der internen Messmethode zeigt sich nachteilig, dass die durchgeführte Prozessüberwachung zu einer erheblichen Verlangsamung der Programmausführung führt. Der Analysevorgang nimmt durchschnittlich 4 Minuten in Anspruch, weshalb umfangreiche quantitative Analysen, wie in sie Teil 1 dieser Arbeit durchgeführt wurden, nicht realisierbar sind.

Kontextbildung

Die Analysemodelle belegen, dass eine umfassende Analyse von Web-Tracking nicht allein auf den Browser begrenzt sein darf. Auch das umliegende System muss berücksichtigt werden. Es wurde gezeigt, dass Systembedingungen kontextbildend in das Verhalten des Browsers eingreifen und infolgedessen nicht unberücksichtigt bleiben dürfen.

Ausblick

Im Ausblick steht eine Erweiterung, die auch interne Abläufe stärker berücksichtigt. Für DisTrack wurde sich für eine HAR-Log basierte Analyse entschieden, um einer Modifikation des Browsers zu entgehen. Seit Mozilla Firefox 47 bietet dieser die vollständige Unterstützung von Remote Debug-

ging[16]. Auf dessen Basis ist eine vollständige Fernsteuerung und Erhebung von internen Browserinformationen möglich. Das Protokoll[17] ist noch nicht vollständig spezifiziert und es stehen keine Programmpakete zur Nutzung in Entwicklungsumgebungen zur Verfügung. Zukünftig kann der Einsatz dieses Protokolls als Ersatz von Selenium und dem WebChecker dienen.

Weitere Browser

Eine weitere Entwicklungsmöglichkeit von DisTrack wäre die Betrachtung weiterer Browser und deren Effekte auf das System. Während der Internet Explorer 8 sich in einer Voruntersuchung [189] als vergleichsweise invasiv zeigte, beschränken sich die Schreibprozesse des Mozilla Firefox auf wenige Systemressourcen, wie in Abschnitt 11.6 gezeigt wurde. So ist die Frage offen, wie andere Browser sich auf das umliegende System auswirken.

16 https://developer.mozilla.org/de/docs/Tools/Remote_Debugging/Debugging_Firefox_Desktop, abgerufen am 22.03.2018.

17 https://searchfox.org/mozilla-central/source/devtools/docs/backend/protocol.md, abgerufen am 22.03.2018.

Schritt 2	Schritt 4
00c.dnscookie.com=>144.217.125.14	00c.dnscookie.com=>144.217.125.13
01c.dnscookie.com=>144.217.125.13	01c.dnscookie.com=>144.217.125.14
02c.dnscookie.com=>144.217.125.13	02c.dnscookie.com=>144.217.125.14
03c.dnscookie.com=>144.217.125.13	03c.dnscookie.com=>144.217.125.13
04c.dnscookie.com=>144.217.125.13	04c.dnscookie.com=>144.217.125.13
05c.dnscookie.com=>144.217.125.13	05c.dnscookie.com=>144.217.125.13
06c.dnscookie.com=>144.217.125.14	06c.dnscookie.com=>144.217.125.13
07c.dnscookie.com=>144.217.125.14	07c.dnscookie.com=>144.217.125.13
08c.dnscookie.com=>144.217.125.14	08c.dnscookie.com=>144.217.125.14
09c.dnscookie.com=>144.217.125.13	09c.dnscookie.com=>144.217.125.13
10c.dnscookie.com=>144.217.125.13	10c.dnscookie.com=>144.217.125.14
11c.dnscookie.com=>144.217.125.13	11c.dnscookie.com=>144.217.125.13
12c.dnscookie.com=>144.217.125.14	12c.dnscookie.com=>144.217.125.14
13c.dnscookie.com=>144.217.125.14	13c.dnscookie.com=>144.217.125.14
14c.dnscookie.com=>144.217.125.13	14c.dnscookie.com=>144.217.125.13
15c.dnscookie.com=>144.217.125.14	15c.dnscookie.com=>144.217.125.14
16c.dnscookie.com=>144.217.125.13	16c.dnscookie.com=>144.217.125.13
17c.dnscookie.com=>144.217.125.14	17c.dnscookie.com=>144.217.125.13
18c.dnscookie.com=>144.217.125.14	18c.dnscookie.com=>144.217.125.13
19c.dnscookie.com=>144.217.125.14	19c.dnscookie.com=>144.217.125.13
20c.dnscookie.com=>144.217.125.14	20c.dnscookie.com=>144.217.125.14
21c.dnscookie.com=>144.217.125.13	21c.dnscookie.com=>144.217.125.13
22c.dnscookie.com=>144.217.125.14	22c.dnscookie.com=>144.217.125.14
23c.dnscookie.com=>144.217.125.14	23c.dnscookie.com=>144.217.125.13
24c.dnscookie.com=>144.217.125.14	24c.dnscookie.com=>144.217.125.13
25c.dnscookie.com=>144.217.125.14	25c.dnscookie.com=>144.217.125.14
26c.dnscookie.com=>144.217.125.14	26c.dnscookie.com=>144.217.125.13
27c.dnscookie.com=>144.217.125.13	27c.dnscookie.com=>144.217.125.14
28c.dnscookie.com=>144.217.125.14	28c.dnscookie.com=>144.217.125.14
29c.dnscookie.com=>144.217.125.13	29c.dnscookie.com=>144.217.125.14
30c.dnscookie.com=>144.217.125.14	30c.dnscookie.com=>144.217.125.13
31c.dnscookie.com=>144.217.125.13	31c.dnscookie.com=>144.217.125.14

Tabelle 11.5: Übersicht der DNS-Abfragen und Antworten. Die Antworten stellen die Bits des Cookies dar: 1474212289 in Schritt 2 und 3125851398 in Schritt 4.

12

FAZIT

Entwicklung

Im ersten Teil dieser Arbeit wurde Web-Tracking quantitativ gemessen und insbesondere dessen historische Entwicklung beleuchtet. Während verwandte Arbeiten die Veränderung von Web-Tracking nur über kurze Zeitabschnitte verglichen haben, beantwortet diese Arbeit die Frage nach einem Gesamtbild der Ausbreitungsentwicklung über einen längeren Zeitraum. Die technische Umsetzung der Messung erfolgte durch Entwurf, Implementierung und Einsatz eines Werkzeugs zur Analyse archivierter Webseiten. Die Messergebnisse wurden durch manuelle Werkzeugtests und durch den Vergleich mit verwandten quantitativen Studien geprüft. Die Daten wurden detailliert ausgewertet und in verschiedenen Visualisierungen aufgearbeitet.

Zunahme

Die vermutete deutliche Zunahme von Web-Tracking konnte belegt werden, was durch einen Anstieg um das sechsfache zwischen den Jahren 2005 und 2015 nachgewiesen wurde. Die statistischen Auswertungen zeigen außerdem, dass in den Jahren 2000 bis 2005 Einbettungen seltener vorgenommen und die Inhalte der Webauftritte von Webseitenanbietern selbst bereitgestellt wurden. Besonders traten Dienste von Google (Alphabet Inc.) und von sozialen Netzwerken in Erscheinung. Neben einer quantitativen Zunahme zeigte sich eine deutliche Entwicklung zu einer monopolisierten Struktur. Im Jahr 2015 wurde beispielsweise eine Überwachung von 78 % der analysierten Webseiten durch die Unternehmen Alphabet Inc. und Facebook Inc.

Probleme

Die Zunahme und Monopolisierung von Tracking führt zu verschiedenartigen Problemen. Das Auskunftsrecht einer Person (Art. 15 DSGVO) erfüllt nur dann seinen Zweck, wenn Nutzern sämtliche erhebenden Stellen bekannt sind. Bei den bis zu 57 gemessenen Einbettungen von Dritten im Jahr 2015 ist die Durchsetzung dieses Rechts kaum realistisch. Weitere Probleme ergeben sich aus nahezu unkontrollierbaren Möglichkeiten der Weitergabe. Stellt ein datenhaltendes Unternehmen seinen Dienst ein, ergibt sich außerdem die Frage nach dem Endverbleib der gesammelten Informationen. Ungeklärt sind auch die technischen und organisatorischen Maßnahmen, die zum Schutz dieser Daten getroffen wurden. Informationen, die durch Datenlecks einer ungesicherten Infrastruktur entwendet wurden, sind keiner Kontrolle mehr zugänglich.

Zukünftige Möglichkeiten

Die Monopolbildung, in Verbindung mit der zunehmenden Nutzung des Internets, führt zu stetig wachsenden Datenmengen. Die Nutzung dieser Daten zur Profilbildung und Selektion personalisierter Werbung ist bekannt und offensichtlich. Durch eine Aufbewahrung der Daten ist auch der Weg für zukünftige Verarbeitungswege offen und die Entwicklungen auf dem Gebiet

der neuronalen Netze stellen immer besser werdende Methoden zur Informationsgewinnung in Aussicht. Infolgedessen sind die vollen Konsequenzen dieser massenhaften Datenerfassung heute noch nicht abschätzbar.

Schutz

Der technische Schutz des Internetnutzers endet mit der Datenerhebung. Ab diesem Zeitpunkt kann ein Schutz nur auf Basis rechtlicher Vorschriften erfolgen. Die neuen Erkenntnisse lassen an der rechtlichen Schutzwirkung zweifeln: Die Ergebnisse der Studie A (Abschnitt 8.1) zeigten unvollständige oder fehlerhafte Datenschutzerklärungen auf 30 von 207 stichprobenartig untersuchten Webauftritten. Unter der Annahme, dass die Informationen nicht bewusst den Webseitenbesuchern vorenthalten wurden, ist die Situation durch fehlende Kenntnisse der Betreiber über die rechtlichen Sorgfaltspflichten und Aufklärungsgebote zu klären. Nutzer und Betreiber müssen an dieser Stelle stärker für Datenschutzaspekte sensibilisiert werden.

Erhebungsformen

Ein effektiver Schutz vor missbräuchlicher Nutzung von personenbeziehbaren Daten ist durch eine Begrenzung der Erhebung möglich. Dies gestaltet sich jedoch schwierig, wenn nicht alle Erhebungspunkte bekannt sind. Die präsentierten Studienergebnisse der Studie B (Abschnitt 8.2) zeigten auch Besuchertracking auf Basis funktionaler Webseitenbestandteile. Durch die zusätzliche Berücksichtigung von Schriftarten, Karten- oder Videodiensten wurde auf 509 von 835 Klinikwebseiten ein Tracking durch Google ermöglicht. Eine rechtliche Betrachtung zeigte auf, dass diese Dienste zukünftig wie typische Trackingdienste behandelt werden müssen und dass Besucher über deren Nutzung aufzuklären sind. Informationen über den gesundheitlichen Zustand werden durch Datenschutzgesetze in besonderer Weise geschützt (Art. 9 DSGVO). Infolgedessen muss Web-Tracking auf Krankenhaus- und Klinikwebseiten besonders kritisch hinterfragt werden.

Vermessung

Die Ausbreitung von Web-Tracking ist ein Teil der zunehmenden und allgegenwärtigen Vermessung, die in alle Lebensbereiche einzieht. Beispiele sind die Erfassung von Vitaldaten durch Fitnesstracker oder die Bewertung der Finanzkraft durch Scoring-Unternehmen. In China wird ein *Social Credit System* erprobt, das über ein Punktesystem die soziale Wertigkeit der Bürger untereinander vergleichbar macht. Besonders bedenklich ist, dass der Score nicht nur vom Nutzer selbst, sondern auch von Behörden, Banken, Arbeitgebern, Vermietern und weiteren einsehbar sein wird.

Resümee

Die Grenzen dieser Vermessung muss Teil des zukünftigen gesellschaftlichen Diskurses sein. Dieser setzt ein Grundverständnis der Situation voraus. Dieses Ziel wurde durch die Messung in dieser Arbeit verfolgt. Die Ergebnisse der veröffentlichten Untersuchungen (Abschnitt 1.5) konnten bereits in den Publikationen von Helm [73], Krieger [95] und Hauschke [71] genutzt werden.

Zukunft

Nach Klärung der historischen Entwicklung und der aktuellen Situation wurde mit *DisTrack* eine Erweiterung bisheriger Messwerkzeuge für die Zukunft geschaffen. Die vorgestellten Modelle belegen, dass der Browserkontext sich nicht allein aus dem Browserprofil ergibt, sondern auch das umliegende System berücksichtigt werden muss. Durch eine Verhaltensbetrachtung des Browsers während dessen Ausführung können darüber hin-

aus Schwachstellen methodisch aufgedeckt werden, die zum Tracking ausgenutzt werden können. Die Schnittmenge von rechtlichen Grundlagen und sich immer weiterentwickelnden technischen Möglichkeiten und Tracking-Werkzeugen erfordert eine stetige Auseinandersetzung mit dem Thema und den aktuellen Technologien. *DisTrack* ist flexibel und kann daher leicht erweitert werden, um den Anforderungen der Zukunft gerecht zu werden.

Ausblick

Weitere Forschungsfragen ergeben sich durch den geänderten gesetzlichen Rahmen. Die eingeführte Datenschutzgrundverordnung (DSGVO) hat weitreichende Änderungen an der gesetzlichen Lage bewirkt. Erste Entwürfe der E-Privacy-Verordnung zeigen mögliche Änderungen, die technischer Unterstützung bedürfen. So steht beispielsweise die verpflichtende Berücksichtigung des Do-Not-Track-Headers zur Diskussion. Möglichkeiten zur Weiterentwicklung ergeben sich beim DisTrack-Framework zur Steigerung der Effizienz und der Entwicklung weiterer Analysemodelle.

Ende

Mit diesem Fazit zur Vergangenheit, Gegenwart und Zukunft findet die vorliegende Arbeit ihren Abschluss.

Teil IV

APPENDIX

A

ABBILDUNGEN

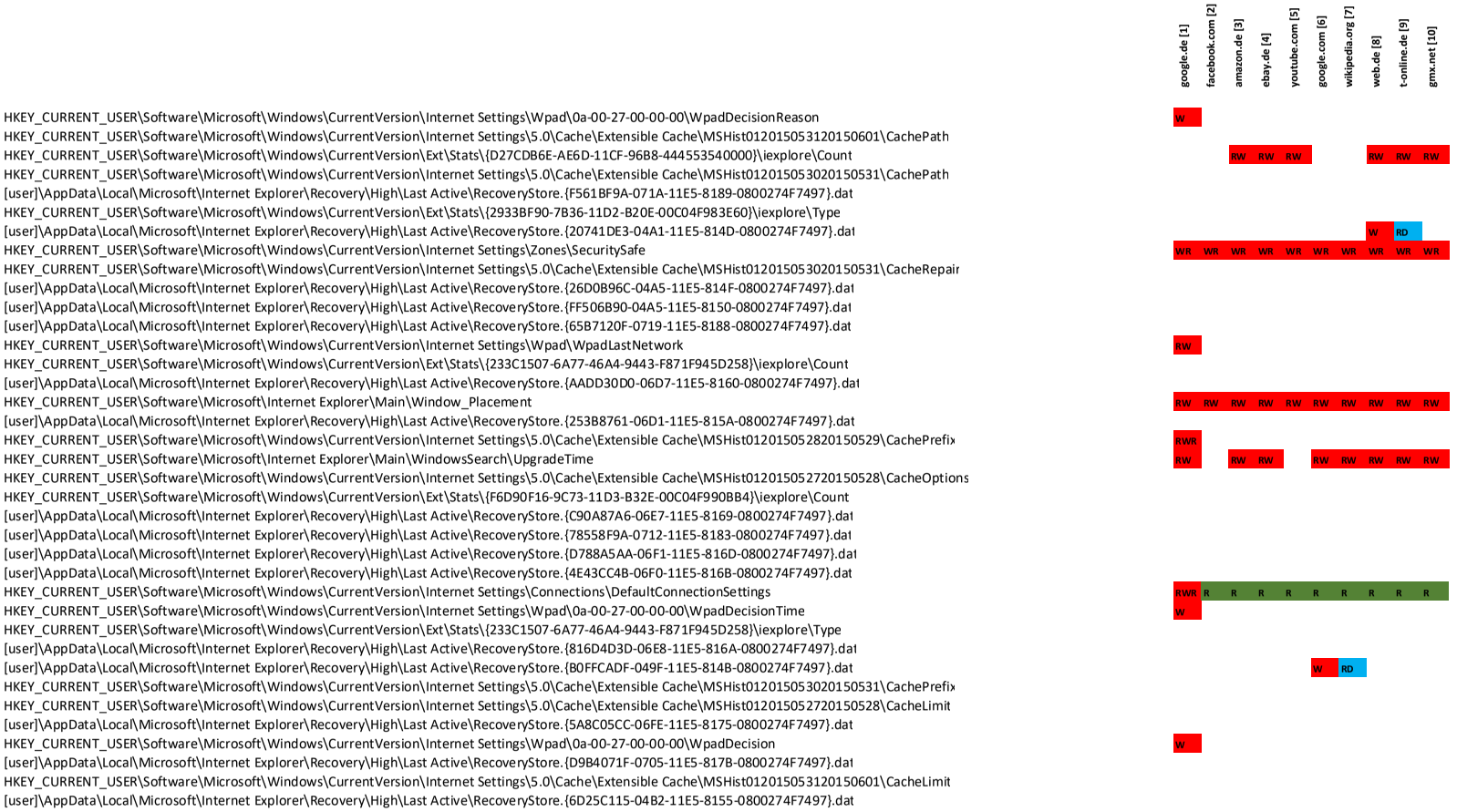

Abbildung A.1: Auszug der Lese- und Schreibzugriffe des Internet Explorers 8 auf die Windows-Registry bei sequenziellem Webseitenabruf. R=Lesen (grün), W=Schreiben (rot), D=Löschen (blau).

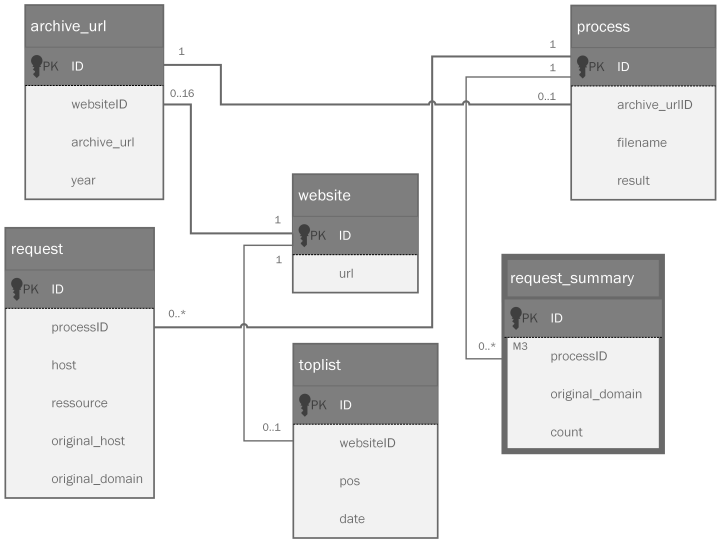

Abbildung A.2: Erweiterung von Abbildung 5.2 um die Hilfstabelle `request_summary` (hervorgehoben).

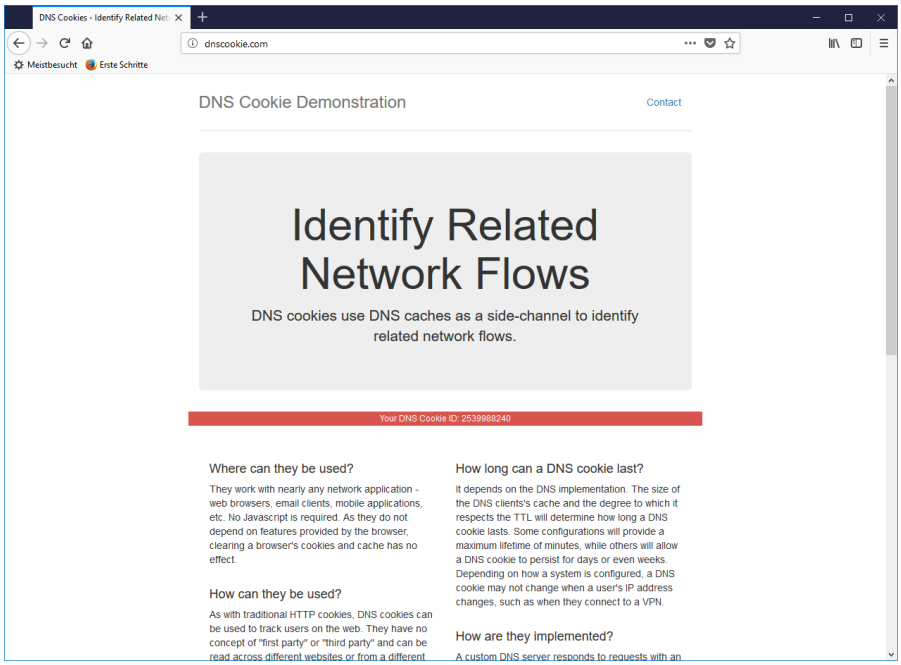

Abbildung A.3: Screenshot der DNS-Cookie-Webseite vom 23.03.2018.

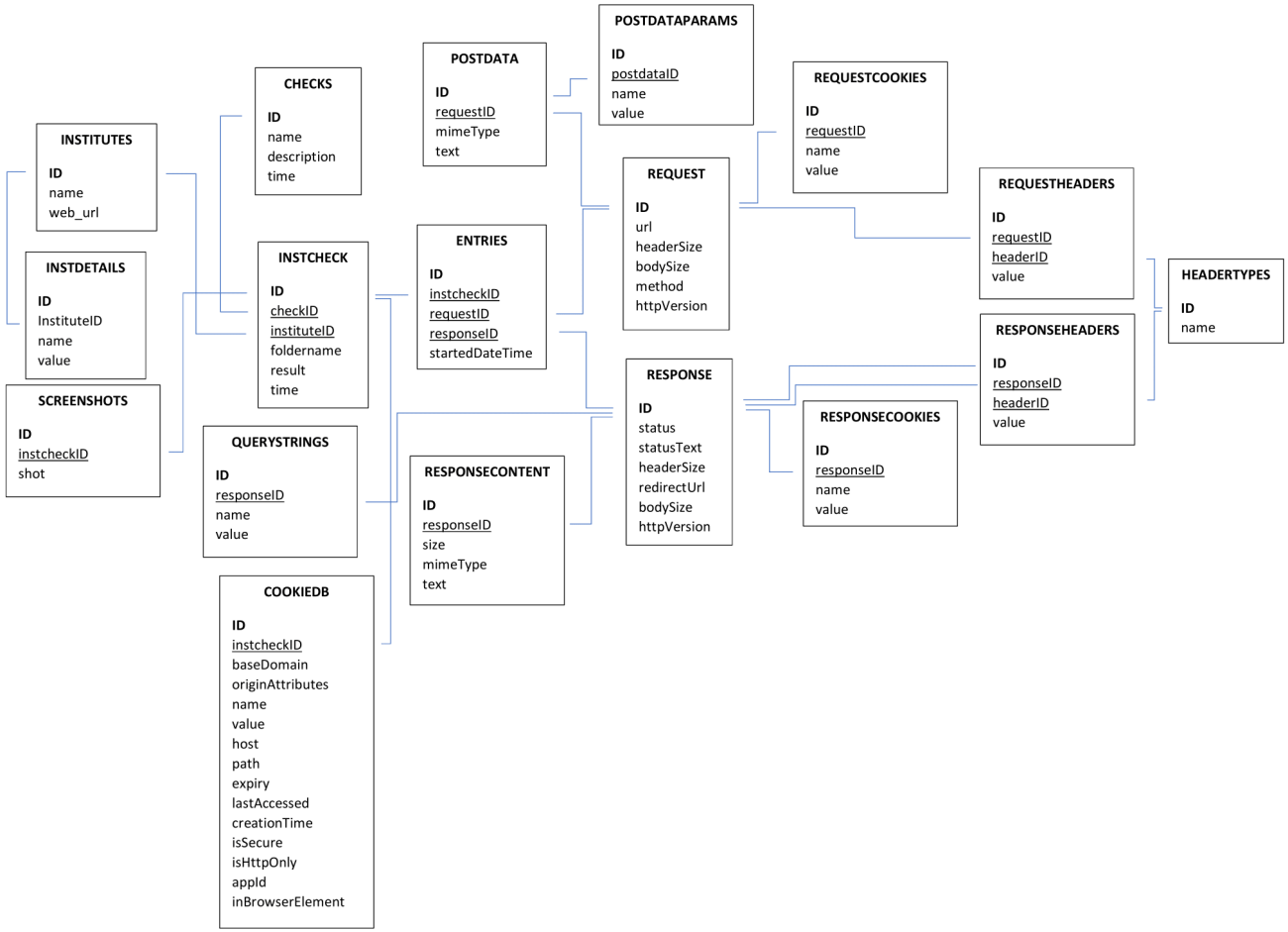

Abbildung A.4: Datenbanklayout bei der Analyse der Krankenhaus- und Klinikwebseiten.

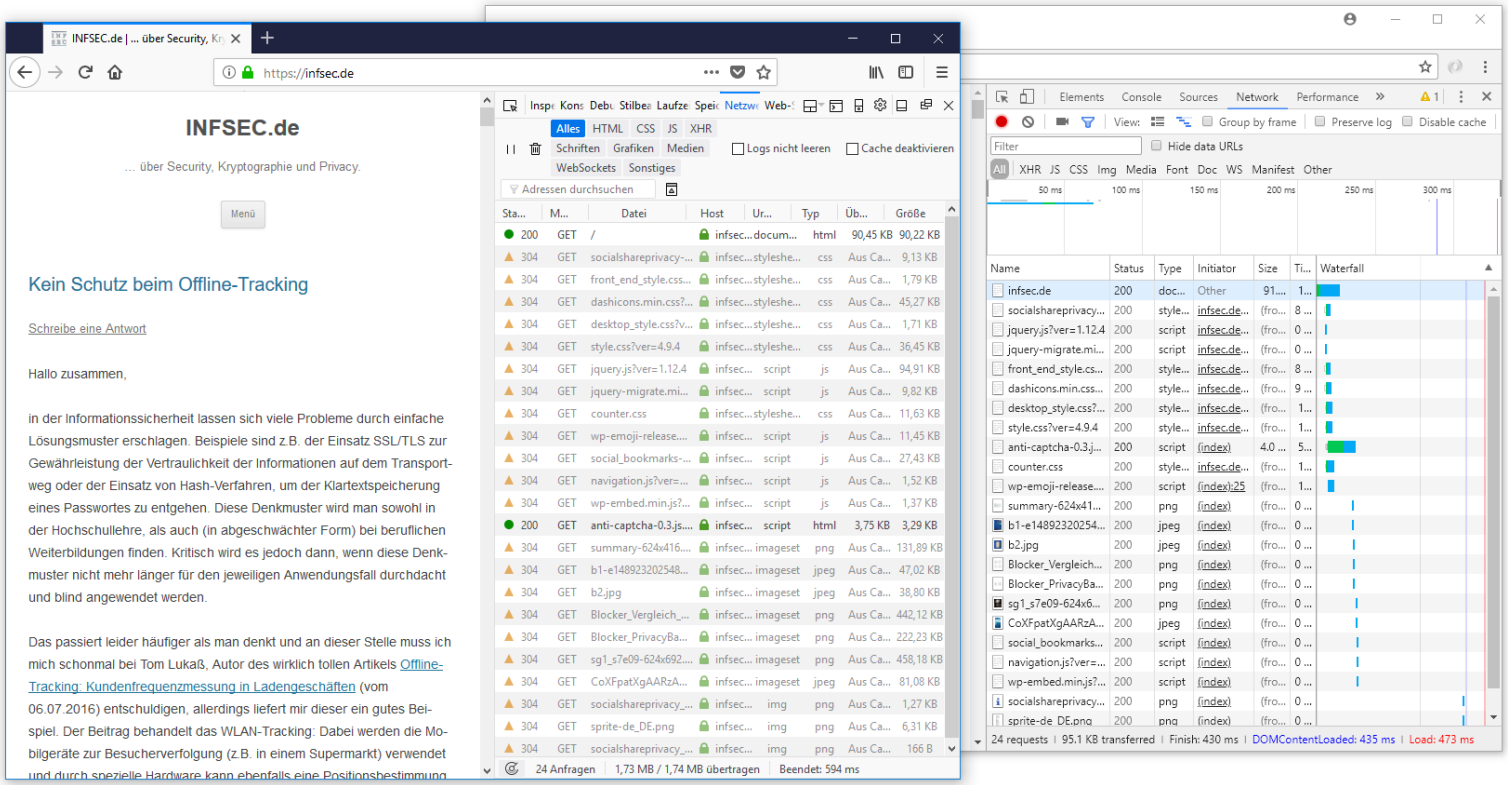

Abbildung A.5: Vergleich der Ergebnisse in den Entwicklerwerkzeuge von Mozilla Firefox (links) und Chrome Browser (rechts) nach Abruf von https://infsec.de.

B

TABELLEN

Nr	Webseite	Datum (Live)	Datum (Archive)	Live-Web	Archive	Übereinstimmend
1	google.com	02.10.17 16:18	02.10.17 00:01:47	1	0	2
2	reddit.com	02.10.17 16:20	02.10.17 00:39:58	7	1	5
3	ebay.com	02.10.17 16:22	02.10.17 05:42:36	0	1	3
4	msn.com	02.10.17 16:24	02.10.17 13:21:34	0	1	2
5	washingtonpost.com	02.10.17 16:27	02.10.17 13:30:06	14	0	9
6	bbc.com	02.10.17 16:29	02.10.17 13:13:03	4	2	6
7	spiegel.de	02.10.17 16:32	02.10.17 14:25:07	23	1	7
8	yahoo.com	02.10.17 16:51	02.10.17 13:35:00	52	0	3
9	wordpress.com	02.10.17 16:54	02.10.17 08:59:44	12	3	13
10	nytimes.com	02.10.17 16:56	02.10.17 13:42:26	40	2	11

Tabelle B.1: Vergleich Anzahl der Drittparteien nur im Live Web, nur im Archived Web und in beiden übereinstimmend.

C

QUELLTEXTE

C.1 PYTHON QUELLTEXTE

```
1
2  if os.path.exists(os.path.join(self.network_path, 'SSLKEYLOGFILE.
       txt')):
3      with open(os.path.join(self.network_path, 'SSLKEYLOGFILE.txt')
           , "r") as data_file:
4          data=data_file.readlines()
5
6      for l in data:
7          if not l.startswith("#"):
8              _, client_random, master_secret = l.split(" ")
9
10             for entry in results.get("tls", {}):
11                 if entry['client_random'] == client_random:
12                     client_random = client_random.decode("hex")
13                     server_random = entry['server_random'].decode
                           ("hex")
14                     master_secret = str(master_secret).replace("\
                           n","").replace("\r","").decode("hex")
15                     tlsmaster[client_random, server_random] =
                           master_secret
```

Quelltext C.1: Modifikation der `processing/network.py` zur Verarbeitung von SSLKEYLOGFILE Schlüsselmaterial.

```
1
2  def start(self, label, task):
3      name = self.db.view_machine_by_label(label).snapshot
4
5      if "source_snapshot" in task.options:
6          name = task.options['source_snapshot']
7
8      with SmartConnection(**self.connect_opts) as conn:
9          vm = self._get_virtual_machine_by_label(conn, label)
10         if vm:
11             self._revert_snapshot(vm, name)
12         else:
13             raise CuckooMachineError(
14                 "Machine %s not found on host" % label
15             )
16
17 def prestop(self, label, task):
18         if "target_snapshot" in task.options:
19
20           time.sleep(5)
21           name = task.options['target_snapshot']
22
23           with SmartConnection(**self.connect_opts) as conn:
24               vm = self._get_virtual_machine_by_label(conn, label
                     )
25               if vm:
26                   self._create_snapshot(vm, name)
```

```
27                 else:
28                     raise CuckooMachineError(
29                         "Machine %s not found on host" % label
30                     )
31
32         if "delete_snapshot" in task.options:
33
34           dels = task.options['delete_snapshot'].split(";")
35           for d in dels:
36               with SmartConnection(**self.connect_opts) as conn:
37                   vm = self._get_virtual_machine_by_label(conn,
                         label)
38                   if vm:
39                       self._delete_snapshot(vm, d)
40                   else:
41                       raise CuckooMachineError(
42                           "Machine %s snapshot not found on host"
                                 % label
43                       )
```

Quelltext C.2: Modifikation der `machinery/vsphere.py` zur Auswahl, Erstellung und Löschung von Snapshots der virtuellen Umgebung.

```
1
2  from lib.common.abstracts import Package
3  from lib.common.results import upload_to_host
4
5  import os
6  import base64
7  import shutil
8  import sys
9  import logging
10
11 log = logging.getLogger(__name__)
12
13 class WebChecker(Package):
14
15     PATHS = [
16         ("HomeDrive", "Python24", "python.exe"),
17         ("HomeDrive", "Python25", "python.exe"),
18         ("HomeDrive", "Python26", "python.exe"),
19         ("HomeDrive", "Python27", "python.exe"),
20         ("HomeDrive", "Python32", "python.exe"),
21         ("HomeDrive", "Python33", "python.exe"),
22         ("HomeDrive", "Python34", "python.exe"),
23     ]
24
25
26     def start(self, url):
27         python = self.get_path("Python")
28         bin_path = os.path.join(self.analyzer.path, "bin")
29
30         path = os.path.join(bin_path,"webcheck.py")
31         b64url = base64.b64encode(url)
32         self.target_path = os.path.join(self.curdir, "
               webc_results")
33
34         arg = [path]
35
36         if "webc_sourceprofile" in self.options:
37             arg = arg + ["--sourceprofile",os.path.join(self.
                  analyzer.path, "files", self.options["
                  webc_sourceprofile"])]
```

```
38          if "webc_targetprofile" in self.options:
39              arg = arg + ["--targetprofile", self.options["
                    webc_targetprofile"]]
40          if "webc_timeout" in self.options:
41              arg = arg + ["--timeout", self.options["webc_timeout"
                    ]]
42          if "webc_waittime" in self.options:
43              arg = arg + ["--waittime", self.options["
                    webc_waittime"]]
44          if "webc_testname" in self.options:
45              arg = arg + ["--testname", self.options["
                    webc_testname"]]
46          if "webc_private" in self.options:
47              arg = arg + ["--private"]
48          if "webc_waitafter" in self.options:
49              arg = arg + ["--waitafter"]
50          if "webc_waitbefore" in self.options:
51              arg = arg + ["--waitbefore"]
52
53          if "webc_b64exportfiles" in self.options:
54              b64expf = base64.b64decode(self.options["
                    webc_b64exportfiles"] )
55              arg = arg + ["--exportfiles", b64expf]
56
57          arg = arg + ["--outputfolder",self.target_path]
58          arg = arg + ["--libpath",os.path.join(self.analyzer.path,
                 "bin", "lib")]
59          arg = arg + ["--base64url", b64url]
60
61          return self.execute(python, args=arg, trigger="file:%s" %
                path)
62
63
64      def finish(self):
65          for f in os.listdir(self.target_path):
66              if os.path.isdir(os.path.join(self.target_path,f)):
67                  for f2 in os.listdir(os.path.join(self.
                        target_path,f)):
68                      upload_to_host(os.path.join(self.target_path,
                            f,f2), os.path.join("webc", f, f2))
69              else:
70                  upload_to_host(os.path.join(self.target_path,f),
                        os.path.join("webc", f))
71
72          shutil.rmtree(self.target_path)
73
74          return True
```

Quelltext C.3: Das webc-Analysemodul zur Durchführung der HAR-Log basierten Webanalyse.

```
1
2   import logging
3   import os
4   import json
5   import base64
6
7   from cuckoo.common.abstracts import Processing
8
9   log = logging.getLogger(__name__)
10
11  class WebChecker(Processing):
12      """Process Webchecker Data"""
13
```

```
14      order = 3
15      key = "webchecker"
16
17
18      def run(self):
19
20          results = {}
21          self.webchecker_path = os.path.join(self.analysis_path, "
                webc")
22
23          webc_data = None
24          try:
25              with open(os.path.join(self.webchecker_path,"result.
                    json")) as data_file:
26                  webc_data = json.load(data_file)
27          except:
28              pass
29
30          if not webc_data:
31              log.warning(
32                  "Unable to find webchecker results"
33                  "skipped."
34              )
35              return results
36
37          results = webc_data
38
39          if 'checks' not in webc_data:
40              log.warning(
41                  "Invalid webchecker results"
42              )
43              return results
44
45          results["har"] = dict()
46          for check_id,cont in webc_data['checks'].items():
47              results['checks'][check_id]['har'] = []
48              for har_file in cont['harfiles']:
49                  har = self._extract_har(os.path.join(self.
                        webchecker_path, check_id, har_file))
50                  results['checks'][check_id]['har'].append(har)
51
52              if 'screenshot' in cont and cont['screenshot'] != "":
53                  cont['screenshot_data'] = self.
                        _extract_screenshot(os.path.join(self.
                        webchecker_path,check_id,cont['screenshot']))
54
55          return results
56
57      def _extract_screenshot(self, path):
58          data = None
59
60          try:
61              with open(path, "rb") as f:
62                  d = f.read()
63
64              if d:
65                  data = base64.b64encode(d)
66          except:
67              pass
68
69          return data
70
71
72      def _extract_har(self, path):
73          har = dict()
74
```

```
75          try:
76              with open(path) as data_file:
77                  har = json.load(data_file)
78          except:
79              pass
80
81          return har
```

Quelltext C.4: Das webc-Verarbeitungsmodul zur Aufbereitung der Daten der webc-Analyseergebnisse.

```
1
2   def report(self, analysis):
3
4           t1,t2,t3,t4 = analysis.get_steps()
5
6           a = self.get_task_report(t2)
7           summary1 = a.behavior.get_firefox_summary().short_summary
8
9           a = self.get_task_report(t4)
10          summary2 = a.behavior.get_firefox_summary().short_summary
11
12          for rw in summary2['file_read']:
13              if rw in summary1['file_created'] and rw not in
                    summary2['file_created']:
14                  h1=t2.get_profile_after().get_hash_file(rw)
15                  h2=t3.get_profile_after().get_hash_file(rw)
16                  dist= str(fuzz.ratio(t2.get_profile_after().
                        get_profile_file(rw), t3.get_profile_after().
                        get_profile_file(rw)))
17                  self._output(rw,h1,h2,dist)
18
19  def _output(self, filename, h1, h2, dist):
20          print "#"*60
21          print filename
22          print "Hash " + ("gleich" if h1==h2 else "ungleich")
23          print "Levenshtein-Distanz: " + dist
```

Quelltext C.5: Auswertung der Messergebnisse nach Ausführung von Quelltext 11.3 (Vollständig).

C.2 SQL-ABFRAGEN

```
1   SELECT Cast(a AS FLOAT)/Cast(b AS FLOAT)
2     FROM (
3       SELECT Count(*) AS a,
4         (
5             SELECT DISTINCT process.websiteid
6             FROM   process
7             JOIN   archive_url
8             ON     (
9                      archive_url.id = process.archive_urlid)
10            AND    process.result=0
11            AND    year = RYEAR) AS b
12      FROM (
13            SELECT DISTINCT archive_url.websiteid AS webid
14            FROM   request_summary
15            JOIN   process
16            ON     (
```

```
17                   request_summary.processid = process.id )
18            JOIN   archive_url
19            ON     (
20                    archive_url.id = process.archive_urlid )
21            AND    year = RYEAR
22            AND    request_summary.original_domain IN
23                 (
24                  SELECT original_domain
25                  FROM (
26                       SELECT original_domain,
27                         Count(*) AS cc
28                       FROM  request_summary
29                       LEFT JOIN process
30                       ON  (
31                            request_summary.processid = process.id )
32                       LEFT JOIN archive_url
33                       ON  (
34                            process.archive_urlid = archive_url.id )
35                       WHERE  archive_url.year = RYEAR
36                       GROUP BY original_domain
37                       ORDER BY cc DESC limit RLIMIT))) )
```

Quelltext C.6: Abfrage zur Abdeckungsanalyse pro Jahr (RYEAR) für die ersten RLIMIT Drittparteien.

```
1  SELECT DISTINCT process.websiteID
2  FROM   request_summary
3         LEFT JOIN process
4                ON ( request_summary.processid = process.id )
5         LEFT JOIN archive_url
6                ON( process.archive_urlid = archive_url.id )
7  WHERE  archive_url.year = RYEAR
8         AND original_domain IN ( 'facebook.com', 'facebook.net', '
              fbcdn.net' )
```

Quelltext C.7: Abfrage bzgl. Anzahl von Facebook-Einbettungen im jeweilign Jahr (RYEAR).

```
1  SELECT DISTINCT process.websiteID
2  FROM   request_summary
3         LEFT JOIN process
4                ON ( request_summary.processid = process.id )
5         LEFT JOIN archive_url
6                ON( process.archive_urlid = archive_url.id )
7  WHERE  archive_url.year = RYEAR
8         AND (original_domain LIKE '%google%' OR original_domain
              LIKE '%doubleclick%')
```

Quelltext C.8: Abfrage bzgl. Anzahl von Google-Einbettungen im jeweilign Jahr (RYEAR).

```
1  SELECT Count(*)
2  FROM   (SELECT DISTINCT archive_url.websiteid
3          FROM   request
4                 LEFT JOIN process
5                        ON ( request.processid = process.id )
6                 LEFT JOIN archive_url
7                        ON ( process.archive_urlid = archive_url.id
                              )
8          WHERE  ( request.original_host = 'apis.google.com'
```

```
9                     OR request.original_host = 'plus.google.com' )
10                  AND archive_url.year = RYEAR
11                  AND process.status = 2)
```

Quelltext C.9: Abfrage bzgl. Anzahl von Google-Plus (apis.google.com) im jeweilign Jahr (RYEAR).

C.3 SONSTIGE QUELLTEXT

```
1  ######################################################################
2  [TP]\cookies.sqlite-wal
3  Hash ungleich
4  Levenshtein-Distanz: 99
5  ######################################################################
6  [TP]\cache2\entries\3DD40E9435F935B585625FCDCE99A47CFD21EF83
7  Hash ungleich
8  Levenshtein-Distanz: 0
9  ######################################################################
10 [TP]\serviceworker.txt
11 Hash ungleich
12 Levenshtein-Distanz: 2
13 ######################################################################
14 [TP]\storage\default\https+++www.washingtonpost.com\cache\caches.
       sqlite
15 Hash ungleich
16 Levenshtein-Distanz: 93
17 ######################################################################
18 [TP]\formhistory.sqlite
19 Hash ungleich
20 Levenshtein-Distanz: 99
21 ######################################################################
22 [TP]\cache2\entries\B2BA4BB78A603ABE794AC399A5EC85ABA283EAE3
23 Hash ungleich
24 Levenshtein-Distanz: 0
25 ######################################################################
26 [TP]\storage\default\https+++www.washingtonpost.com\idb
       \12183338011.sqlite
27 Hash gleich
28 Levenshtein-Distanz: 100
29 ######################################################################
30 [TP]\cache2\entries\26DA03BED32A114E95DB8881FB4D6A7EFBEB466B
31 Hash ungleich
32 Levenshtein-Distanz: 0
33 ######################################################################
34 [TP]\storage\default\https+++www.washingtonpost.com\idb\301792106
       ttes.sqlite
35 Hash gleich
36 Levenshtein-Distanz: 100
```

Quelltext C.10: Vollständiges Ergebnisausgabe aus Abschnitt 11.4.2

```
- summary
        - file_deleted
                0       "C:\\Users\\lab\\AppData\...ootlss--cans.sqlite-wal"
                1       "C:\\Users\\lab\\AppData\...missions.sqlite-journal"
                2       "C:\\Users\\lab\\AppData\...ss--cans.sqlite-journal"
                3       "C:\\Users\\lab\\AppData\...Yv\\favicons.sqlite-shm"
                4       "C:\\Users\\lab\\AppData\...\\places.sqlite-journal"
                5       "C:\\Users\\lab\\AppData\...ootlss--cans.sqlite-shm"
                6       "C:\\Users\\lab\\AppData\...5piupsah.sqlite-journal"
                7       "C:\\Users\\lab\\AppData\...nt-prefs.sqlite-journal"
                8       "C:\\Users\\lab\\AppData\...63365piupsah.sqlite-wal"
                9       "C:\\Users\\lab\\AppData\...favicons.sqlite-journal"
                10      "C:\\Users\\lab\\AppData\...\cookies.sqlite-journal"
                11      "C:\\Users\\lab\\AppData\...63365piupsah.sqlite-shm"
                12      "C:\\Users\\lab\\AppData\...\storage.sqlite-journal"
                13      "C:\\Users\\lab\\AppData\...7XYv\\d3d11layers.guard"
                14      "C:\\Users\\lab\\AppData\...Yv\\favicons.sqlite-wal"
                15      "C:\\Users\\lab\\AppData\...ppsstore.sqlite-journal"
        + file_created                  [...]
        + file_recreated                [...]
        + directory_created             [...]
        + dll_loaded                    [...]
        + file_opened                   [...]
        + file_copied                   [...]
        + regkey_opened                 [...]
        + file_moved                    [...]
        + file_written                  [...]
        + resolves_host                 [...]
        + connects_ip                   [...]
        + directory_removed             [...]
        + file_exists                   [...]
        + file_failed                   [...]
        + wmi_query                     [...]
        + guid                          [...]
        + file_read                     [...]
        + regkey_read                   [...]
        + directory_enumerated          [...]
```

Quelltext C.11: Auszug des DisTrack-Ergebnisberichts.

D

TESTMENGEN

D.1 MANUELLE PRÜFUNG ARCHIVIERTER WEBSEITEN

- http://web.archive.org/web/20070629174626/http://www.poczta-polska.pl:80/
- http://web.archive.org/web/20040630220850/http://www.uclm.es:80/
- http://web.archive.org/web/20090618025454/http://www.smhi.se:80/?
- http://web.archive.org/web/20110629104130/http://www.quechoisir.org:80/
- http://web.archive.org/web/20130721165833/http://xht888.com/
- http://web.archive.org/web/20080629191142/http://www.google.com.ng:80/
- http://web.archive.org/web/20120908042741/http://www.evidus.com:80/
- http://web.archive.org/web/20010701043735/http://www.jibjab.com:80/
- http://web.archive.org/web/20080701030634/http://www.imagegateway.net:80/
- http://web.archive.org/web/20070705021427/http://www.esquire.com:80/
- http://web.archive.org/web/20110628003924/http://paultan.org/
- http://web.archive.org/web/20060701012410/http://www.hemmings.com:80/
- http://web.archive.org/web/20070108120113/http://www.bjjs.gov.cn:80/
- http://web.archive.org/web/20040704072205/http://www.erstebank.hu:80/
- http://web.archive.org/web/20120701040705/http://www.animenewsnetwork.com:80/
- http://web.archive.org/web/20140625072830/http://www.scrapbook.com/
- http://web.archive.org/web/20050701082940/http://www.readwritethink.org:80/
- http://web.archive.org/web/20040701064710/http://www.sba.gov:80/
- http://web.archive.org/web/20070404003219/http://www.smarthome.com:80/
- http://web.archive.org/web/20131115235111/http://www.crocs.com/
- http://web.archive.org/web/20140630194044/http://www.komputerswiat.pl/
- http://web.archive.org/web/20100702074223/http://www.bestplaces.net/
- http://web.archive.org/web/20090524190702/http://www.nntt.org:80/
- http://web.archive.org/web/20040701050335/http://info.com:80/
- http://web.archive.org/web/20050630234944/http://westmarine.com:80/
- http://web.archive.org/web/20050701010948/http://www.thelotter.com:80/
- http://web.archive.org/web/20150628130641/http://www.sejm.gov.pl/
- http://web.archive.org/web/20070701012326/http://www.google.ch./
- http://web.archive.org/web/20130627221134/http://www.destructoid.com:80/?
- http://web.archive.org/web/20040630161224/http://www.oper.ru:80/
- http://web.archive.org/web/20030623144633/http://www.freep.com:80/
- http://web.archive.org/web/20140628020422/http://joomlaforum.ru/
- http://web.archive.org/web/20030620234838/http://euroauto.ru:80/
- http://web.archive.org/web/20080701050552/http://www.polldaddy.com:80/
- http://web.archive.org/web/20080701100658/http://www.komputronik.pl:80/?
- http://web.archive.org/web/20120630140212/http://www.ouest-france.fr:80/
- http://web.archive.org/web/20090701184307/http://www.autocar.co.uk:80/
- http://web.archive.org/web/20060701033757/http://allnurses.com:80/
- http://web.archive.org/web/20090706210249/http://www.chevrolet.com:80/
- http://web.archive.org/web/20010516001015/http://www.helios.pl:80/
- http://web.archive.org/web/20110629223801/http://www.popularmechanics.com:80/
- http://web.archive.org/web/20080621113212/http://www.assafir.com:80/
- http://web.archive.org/web/20030611235216/http://www.yarn.com:80/
- http://web.archive.org/web/20130706045416/http://www.instapage.com:80/
- http://web.archive.org/web/20120629030450/http://www.zhulong.com:80/
- http://web.archive.org/web/20100501062002/http://whatismyipaddress.com:80/
- http://web.archive.org/web/20080710124621/http://dealnews.com/
- http://web.archive.org/web/20110630140855/http://www.avocatnet.ro:80/
- http://web.archive.org/web/20040701024345/http://www.naviance.com:80/
- http://web.archive.org/web/20150629064641/http://www.yam.com:80/

D.2 WEBSEITEN VON HOCHSCHULEN

Evangelische Fachhochschule Rheinland-Westfalen-Lippe (Bochum) http://efh-bochum.de, Macromedia Hochschule fuer Medien und Kommunikation (Muenchen) http://www.macromedia-fachhochschule.de, Private Fachhochschule Goettingen https://www.pfh.de, Hochschule fuer nachhaltige Entwicklung Eberswalde http://www.hnee.de, Hochschule Neubrandenburg http://www.hs-nb.de, Katholische Stiftungsfachhochschule Muenchen http://www.ksfh.de, Technische Fachhochschule Georg Agricola (Bochum) https://www.tfh-bochum.de, Hochschule fuer angewandtes Management (Erding) http://www.fham.de, Hochschule Hamm-Lippstadt http://www.hshl.de, **HafenCity Universitaet Hamburg http://www.hcu-hamburg.de**, Hochschule Biberach http://www.hochschule-biberach.de, Fachhochschule Stralsund http://www.fh-stralsund.de, **Tieraerztliche Hochschule Hannover http://www.tiho-hannover.de**, Fachhochschule Nordhausen http://www.fh-nordhausen.de, Fachhochschule Bingen http://www.fh-bingen.de, Hochschule Ansbach http://www.hs-ansbach.de, Helmut-Schmidt-Universitaet (Hamburg) http://www.hsu-hh.de, Paedagogische Hochschule Schwaebisch Gmuend http://www.ph-gmuend.de, Hessische Hochschule fuer Polizei und Verwaltung (Wiesbaden) https://hfpv.hessen.de, Hochschule Merseburg http://www.hs-merseburg.de, SRH Hochschule Heidelberg http://www.hochschule-heidelberg.de, Fachhochschule Schmalkalden http://www.fh-schmalkalden.de, Hochschule Aschaffenburg http://www.h-ab.de, Fachhochschule Brandenburg http://fh-brandenburg.de, Hochschule Albstadt-Sigmaringen http://www.hs-albsig.de, Universitaet der Bundeswehr Muenchen http://www.unibw.de, Hochschule Hof http://www.hof-university.de, Alice Salomon Hochschule Berlin http://www.ash-berlin.eu, Hochschule Bremerhaven http://www.hs-bremerhaven.de, Hochschule fuer angewandte Wissenschaften Neu-Ulm https://hs-neu-ulm.de, Fachhochschule Worms http://www.hs-worms.de, Hochschule Harz (Wernigerode und Halberstadt) https://www.hs-harz.de, Medizinische Hochschule Hannover http://www.mh-hannover.de, **Paedagogische Hochschule Weingarten http://www.ph-weingarten.de**, Fachhochschule Potsdam http://www.fh-potsdam.de, **Hochschule Ravensburg-Weingarten http://www.hs-weingarten.de**, Ostbayerische Technische Hochschule Amberg-Weiden http://www.oth-aw.de, Hochschule Zittau/Goerlitz http://www.hszg.de, Universitaet zu Luebeck http://www.uni-luebeck.de, Deutsche Hochschule fuer Praevention und Gesundheitsmanagement (Saarbruecken) http://www.uni-saarland.de/, Universitaet der Kuenste Berlin http://www.udk-berlin.de, Diploma Hochschule (Bad Sooden-Allendorf) http://diploma.de, Universitaet Vechta http://www.uni-vechta.de, **Paedagogische Hochschule Karlsruhe http://www.ph-karlsruhe.de**, Hochschule fuer Technik Stuttgart http://www.hft-stuttgart.de, Fachhochschule fuer oeffentliche Verwaltung und Rechtspflege in Bayern (Muenchen) http://www.fhvr.bayern.de, Hochschule Ulm http://www.hs-ulm.de, Katholische Hochschule Nordrhein-Westfalen (Koeln, Muenster, Paderborn, Aachen) http://www.katho-nrw.de, Fachhochschule Flensburg http://www.fh-flensburg.de, Hochschule Rosenheim http://www.fh-rosenheim.de, Hochschule der Medien (Stuttgart) http://www.hdm-stuttgart.de, Hochschule Offenburg http://fh-offenburg.de, **Technische Hochschule Wildau (FH) http://www.th-wildau.de**, Bauhaus-Universitaet Weimar http://www.uni-weimar.de, **Hochschule Landshut https://www.haw-landshut.de**, Hochschule Ludwigshafen am Rhein http://www.hs-lu.de, **Hochschule Emden/Leer http://www.hs-emden-leer.de**, Fachhochschule Luebeck http://www.fh-luebeck.de, Technische Hochschule Ingolstadt http://www.thi.de, Hochschule fuer angewandte Wissenschaften Coburg http://www.hs-coburg.de, Paedagogische Hochschule Heidelberg http://www.ph-heidelberg.de, Hochschule Konstanz Technik, Wirtschaft und Gestaltung http://www.htwg-konstanz.de, Hochschule fuer Wirtschaft und Umwelt Nuertingen-Geislingen http://www.hfwu.de, Fachhochschule Erfurt http://www.fh-erfurt.de, Europa-Universitaet Flensburg https://www.uni-flensburg.de, Deutsche Sporthochschule Koeln http://www.dshs-koeln.de, Ernst-Abbe-Fachhochschule Jena http://www.eah-jena.de, Rheinische Fachhochschule Koeln http://www.rfh-koeln.de, Hochschule Anhalt (Bernburg, Dessau und Koethen) http://www.hs-anhalt.de, Technische Hochschule Deggendorf https://www.th-deg.de, Hochschule Mainz https://www.hs-mainz.de, Hochschule fuer angewandte Wissenschaften Kempten http://www.hochschule-kempten.de, **Technische Universitaet Clausthal http://www.tu-clausthal.de**, Westsaechsische Hochschule Zwickau http://www.fh-zwickau.de, Katholische Universitaet Eichstaett-Ingolstadt http://www.ku-eichstaett.de, Hochschule Aalen https://www.hs-aalen.de, Hochschule Rhein-Waal (Kleve, Kamp-Lintfort) http://www.hochschule-rhein-waal.de, Paedagogische Hochschule Freiburg https://www.ph-freiburg.de, Hochschule Reutlingen http://www.reutlingen-university.de, Hochschule Mannheim http://www.hs-mannheim.de, HAWK Hochschule Hildesheim/Holzminden/Goettingen http://www.hawk-hhg.de, AKAD Bildungsgesellschaft (Stuttgart) https://www.akad.de, Hochschule fuer Technik und Wirtschaft Dresden http://www.htw-dresden.de, Hochschule Augsburg http://www.hs-augsburg.de, Hochschule Pforzheim http://www.hs-pforzheim.de, Technische Universitaet Bergakademie Freiberg http://tu-freiberg.de, Hochschule fuer Technik und Wirtschaft des Saarlandes (Saarbruecken) http://www.htwsaar.de, Universitaet Erfurt http://www.uni-erfurt.de, Paedagogische Hochschule Ludwigsburg https://www.ph-ludwigsburg.de, Fachhochschule Kaiserslautern http://fh-kl.de, Europaeische Fernhochschule Hamburg http://www.euro-fh.de, Hochschule Weihenstephan-Triesdorf http://www.hswt.de, Steinbeis-Hochschule Berlin http://www.steinbeis.de, Hochschule Fresenius (Idstein) http://www.hs-fresenius.de, Wilhelm Buechner Hochschule (Pfungstadt) http://www.wb-fernstudium.de, Hochschule Furtwangen http://www.hs-furtwangen.de, Hochschule Mittweida http://www.hs-mittweida.de, Hochschule Esslingen http://www.hs-esslingen.de, Hochschule Bochum http://www.hochschule-bochum.de, Universitaet Hildesheim http://www.uni-hildesheim.de, Hochschule fuer Technik, Wirtschaft und Kultur Leipzig http://www.htwk-leipzig.de, Hochschule Ostwestfalen-Lippe (Lemgo) https://www.hs-owl.de, Technische Universitaet Hamburg-Harburg http://www.tuhh.de, **Jade Hochschule (Wilhelmshaven, Oldenburg, Elsfleth) http://www.jade-hs.de**, Fachhochschule Kiel http://www.fh-kiel.de, **Hochschule Fulda http://www.fh-fulda.de**, Hochschule Magdeburg-Stendal https://www.hs-magdeburg.de, Europa-Universitaet Viadrina (Frankfurt (Oder)) http://www.europa-uni.de, Fachhochschule fuer oeffentliche Verwaltung Nordrhein-Westfalen (Gelsenkirchen) https://www.fhoev.nrw.de, Technische Universitaet Ilmenau http://www.tu-ilmenau.de, Hochschule Bonn-Rhein-Sieg https://www.h-brs.de, Hochschule Trier http://www.hochschule-trier.de, Hochschule Wismar http://www.hs-wismar.de, Hochschule Karlsruhe Technik und Wirtschaft http://www.hs-karlsruhe.de, Leuphana Universitaet Lueneburg http://www.leuphana.de, Hochschule Heilbronn https://www.hs-heilbronn.de, Hochschule Koblenz http://www.hs-koblenz.de, Westfaelische Hochschule Gelsenkirchen Bocholt Recklinghausen http://www.w-hs.de, Hochschule Hannover http://www.hs-hannover.de, Fachhochschule Duesseldorf http://www.hs-duesseldorf.de, Fachhochschule Bielefeld http://www.fh-bielefeld.de, Hochschule fuer angewandte Wissenschaften Wuerzburg-Schweinfurt http://www.fhws.de, Hochschule Bremen http://www.hs-bremen.de, Universitaet Hohenheim (Stuttgart) https://www.uni-hohenheim.de, HFH Hamburger Fern-Hochschule http://www.hamburger-fh.de, Universitaet Ulm http://www.uni-ulm.de, Hochschule fuer Wirtschaft und Recht Berlin http://www.hwr-berlin.de, Universitaet Passau http://www.uni-passau.de, Brandenburgische Technische Universitaet Cottbus-Senftenberg http://www.b-tu.de, Hochschule RheinMain (Wiesbaden und Ruesselsheim) http://www.hs-rm.de, Ostbayerische Technische Hochschule Regensburg https://www.oth-regensburg.de, Fachhochschule Frankfurt am Main http://www.frankfurt-university.de, **Universitaet Konstanz http://www.uni-konstanz.de**, Ostfalia Hochschule fuer angewandte Wissenschaften (Wolfenbuettel, Salzgitter, Wolfsburg, Suderberg) http://www.ostfalia.de, FH Aachen http://www.fh-aachen.de, Fachhochschule Dortmund http://www.fh-dortmund.de, Beuth Hochschule fuer Technik Berlin http://www.beuth-hochschule.de, Technische Universitaet Chemnitz https://www.tu-chemnitz.de, Universitaet Osnabrueck http://www.uni-osnabrueck.de, Universitaet Bayreuth http://www.uni-bayreuth.de, Carl von Ossietzky Universitaet Oldenburg http://www.uni-oldenburg.de, Hochschule Osnabrueck http://www.hs-osnabrueck.de, Fachhochschule Muenster https://www.fh-muenster.de, Fachhochschule Suedwestfalen (Iserlohn) http://www4.fh-swf.de, Ernst-Moritz-Arndt-Universitaet Greifswald http://uni-greifswald.de, Universitaet Mannheim http://www.uni-mannheim.de, Otto-Friedrich-Universitaet Bamberg http://www.uni-bamberg.de, Technische Hochschule Nuernberg Georg Simon Ohm http://www.th-nuernberg.de, Hochschule Niederrhein (Krefeld/Moenchengladbach) http://www.hs-niederrhein.de, **Hochschule fuer Technik und Wirtschaft Berlin http://www.htw-berlin.de**, Technische Universitaet Kaiserslautern https://www.uni-kl.de, Technische Hochschule Mittelhessen (Giessen, Friedberg, Wetzlar) http://www.thm.de, Otto-von-Guericke-Universitaet Magdeburg http://www.uni-magdeburg.de, Universitaet Koblenz-Landau http://www.uni-koblenz-landau.de, Hochschule Darmstadt https://h-da.de, **Universitaet Rostock http://www.uni-rostock.de**, Universitaet Trier http://www.uni-trier.de, Hochschule fuer Angewandte Wissenschaften Hamburg http://www.haw-hamburg.de, Technische Universitaet Braunschweig https://www.tu-braunschweig.de, Hochschule fuer angewandte Wissenschaften Muenchen http://www.hm.edu, Universitaet des Saarlandes (Saarbruecken, Homburg) http://www.uni-saarland.de, Universitaet Siegen http://www.uni-siegen.de, Universitaet Bremen http://www.uni-bremen.de, Universitaet Paderborn http://www.uni-paderborn.de, Friedrich-Schiller-Universitaet Jena http://www.uni-jena.de, Universitaet Bielefeld http://www.uni-bielefeld.de, Universitaet Augsburg http://www.uni-augsburg.de, Universitaet Potsdam http://www.uni-potsdam.de, Bergische Universitaet Wuppertal http://www.uni-wuppertal.de, **Martin-Luther-Universitaet Halle-Wittenberg http://www.uni-halle.de**, Universitaet Regensburg http://www.uni-regensburg.de, FOM Hochschule (Essen) https://www.fom.de, Fachhochschule Koeln https://www.fh-koeln.de, Universitaet Kassel http://www.uni-kassel.de, Gottfried Wilhelm Leibniz Universitaet Hannover http://www.uni-hannover.de, Christian-Albrechts-Universitaet zu Kiel http://www.uni-kiel.de,

Julius-Maximilians-Universitaet Wuerzburg http://www.uni-wuerzburg.de, Karlsruher Institut fuer Technologie http://www.kit.edu, **Albert-Ludwigs-Universitaet Freiburg http://www.uni-freiburg.de**, Technische Universitaet Darmstadt http://www.tu-darmstadt.de, Philipps-Universitaet Marburg http://www.uni-marburg.de, Justus-Liebig-Universitaet Giessen http://www.uni-giessen.de, Universitaet Leipzig http://www.uni-leipzig.de, Universitaet Stuttgart http://www.uni-stuttgart.de, Georg-August-Universitaet Goettingen http://www.uni-goettingen.de, **Eberhard Karls Universitaet Tuebingen http://www.uni-tuebingen.de**, Heinrich-Heine-Universitaet Duesseldorf http://www.uni-duesseldorf.de, Rheinische Friedrich-Wilhelms-Universitaet Bonn http://www3.uni-bonn.de, Ruprecht-Karls-Universitaet Heidelberg http://www.uni-heidelberg.de, Technische Universitaet Dortmund http://www.tu-dortmund.de, Technische Universitaet Berlin http://www.tu-berlin.de, Humboldt-Universitaet zu Berlin http://www.hu-berlin.de, Duale Hochschule Baden-Wuerttemberg (Stuttgart) http://www.dhbw-stuttgart.de, Freie Universitaet Berlin http://www.fu-berlin.de, Johannes Gutenberg-Universitaet Mainz http://www.uni-mainz.de, Technische Universitaet Dresden http://tu-dresden.de, Technische Universitaet Muenchen http://www.tum.de, Friedrich-Alexander-Universitaet Erlangen-Nuernberg http://www.fau.eu, Universitaet Hamburg http://www.uni-hamburg.de, **Universitaet Duisburg-Essen https://www.uni-due.de**, RWTH Aachen http://www.rwth-aachen.de, **Westfaelische Wilhelms-Universitaet (Muenster) http://www.uni-muenster.de**, Ruhr-Universitaet Bochum http://www.ruhr-uni-bochum.de, **Johann Wolfgang Goethe-Universitaet Frankfurt am Main http://www.goethe-university-frankfurt.de**, Universitaet zu Koeln http://www.uni-koeln.de, Ludwig-Maximilians-Universitaet Muenchen https://www.uni-muenchen.de, Fernuniversitaet in Hagen http://www.fernuni-hagen.de

D.3 WEBSEITEN VON KRANKENHÄUSER UND KLINIKEN

http://www.asklepios.com/gauting, https://www.lungenklinik-hemer.de, http://www.pgdiakonie.de/evangelische-lungenklinik-berlin/, http://www.helios-kliniken.de/klinik/berlin-zehlendorf.html, http://www.rbk.de/standorte/klinik-schillerhoehe.html, http://klinikumchemnitz.de, http://www.ruhrlandklinik.de/, http://www.lungenclinic.de, http://www.waldklinikumgera.de, http://www.florence-nightingale-krankenhaus.de, http://www.evk-herne.de, http://www.klinikum-nuernberg.de, http://www.klinik-loewenstein.de, http://www.zentralklinik.de, http://www.fachkrankenhaus-coswig.de/, http://www.johanniter-treuenbrietzen.de, http://www.kliniken-essen-mitte.de/, http://www.rkk-stuttgart.de/home.html, http://www.lungenklinik-lostau.de, http://www.uk-essen.de, http://www.uks.eu, http://www.drk-kliniken-berlin.de/Mitte, https://www.klinikum-muenchen.de/bogenhausen/, http://www.ruppiner-kliniken.de/, http://rangauklinik.de, http://www.clemenshospital.de, http://www.vivantes.de/, http://www.klinikum-bremen-ost.de, http://www.barmherzige-regensburg.de, http://www.maerkische-kliniken.de, http://www.ekweende.de, http://www.uniklinik-freiburg.de, http://www.heinrich-braun-klinikum.de, https://www.asklepios.de/hamburg/harburg/, http://www.vincentius-kliniken.de, http://www.bezirksklinikum-obermain.de, http://www.krankenhaus-nordwest.de/, http://www.charite.de, http://www.gesundheit-nordhessen.de, http://www.kkm-mainz.de, http://www.evkln.de/startseite/johanniter-krankenhaus-oberhausen/, http://www.klinikum-augsburg.de, http://www.pius-hospital.de, http://www.mariahilf.de, http://www.klinikum-braunschweig.de, http://www.tzbu.de, http://www.klinikum.uni-muenchen.de, http://www.klinik-bethanien.de, http://www.sanktgeorg.de/home.html, http://www.lungenklinik-ballenstedt.de, http://www.kkel.de/standorte/st-josef-hospital/ueber-uns/, http://www.ctk.de, http://www.bergmannsheil-buer.de, http://www.niels-stensen-kliniken.de, http://www.klinikumdo.de, http://www.kgu.de, http://www.mkh-soest.de, http://www.klinikum-stuttgart.de/ueber-uns/startseite/katharinenhospital/, http://www.marien-hospital.de, http://www.diako-online.de, http://www.fachkliniken-wangen.de, http://www.malteser-krankenhaus-bonn.de, http://www.uksh.de, http://www.bk-paderborn.de, http://www.med.uni-magdeburg.de/, http://www.mathias-stiftung.de, http://www.ukaachen.de, http://www.martha-maria.de/krankenhaus-halle.php, http://www.bethesda.de, http://www.ukw.de, http://www.bk-trier.de, http://www.krh.eu/klinikum/SOH/Seiten/default.aspx, http://www.klinikum-bremerhaven.de, http://www.marienhospital-stuttgart.de, http://www.klinikum-lev.de, http://www.klinikum-bayreuth.de, http://kk-km.de, http://www.augusta-bochum.de/, http://www.uniklinik-ulm.de/, http://www.uniklinikum-jena.de, http://www.ukr.de, http://www.westpfalz-klinikum.de, http://www.klinikum-lippe.de, http://www.sanderbusch.de, http://www.klilu.de, http://www.marienhospital-hamm.de, http://www.havelhoehe.de, http://www.klinik-waldhof.de, http://www.diakoniekrankenhaus-halle.de/kliniken/, http://www.glg-mbh.de, http://www.uniklinik-leipzig.de, http://www.bergmannsheil.de, http://www.medizin.uni-tuebingen.de, http://www.klinikum-passau.de, http://www.klinikumevb.de, http://w2.umm.de, http://www.malteser-franziskus.de/, http://www.uk-erlangen.de, http://www.med.uni-rostock.de, http://www.muehlenkreiskliniken.de/johannes-wesling-klinikum-minden/jwk-minden/herzlich-willkommen.html, http://www.klinikum-darmstadt.de/, http://www.krankenhaus-werne.de, http://www.zentralklinikum-suhl.de, http://www.klinikum-os.de, http://krankenhaus-dueren.de, http://www.mh-hannover.de, https://www.kk-essen.de, http://www.bernward-khs.de/, http://www.uke.de, http://www.evangelische-kliniken-bonn.de/, http://www.medizin.uni-halle.de, http://www.st-vincenz.de/, http://www.ukb.uni-bonn.de, http://klinik-oeschelbronn.de, http://www.klinikumbielefeld.de/, http://www.sana-oh.de, http://www.mutterhaus.de, http://www.kreisklinikum-siegen.de/, http://www.dbknb.de, http://klinikum-luenen.de/startseite/, http://www.medizin.uni-greifswald.de, http://www.prohomine.de, http://www.klinikum-esslingen.de, http://malteser-stanna.de, http://www.petrus-krankenhaus-wuppertal.de/, http://krankenhaus-halle-saale.de, http://www.malteser-sthildegardis.de, http://www.klinikum-fuerth.de, http://www.universitaetsmedizin-goettingen.de, http://www.kkh-glauchau.de, http://www.caritasklinikum.de, http://www.klinikum-mittelbaden.de, http://www.akh-viersen.de, http://www.akh-hagen.de/, http://www.lahn-dill-kliniken.de, http://www.klinikum-westfalen.de, http://koblenz.bwkrankenhaus.de, http://www.klinikum.uni-muenster.de, http://www.krankenhaus-mol.de, http://www.st-marien-hospital.de, http://www.akh-celle.de, http://www.klinikum-hanau.de, http://romed-kliniken.de, http://www.sana-klinikum-hof.de, http://www.marienkrankenhaus.com, http://www.klinikum-dessau.de/, http://www.klinikum-bochum.de, http://www.diako-bremen.de, http://www.unimedizin-mainz.de/, http://www.uniklinikum-dresden.de, http://www.krh-nk.de, http://www.hochtaunus-kliniken.de, http://www.missioklinik.de, http://www.klinikum-neumarkt.de, http://www.ameos.eu/bernburg, http://www.klinikum-amberg.de, http://www.sbk-vs.de/, http://www.marienhospital-herne.de/, http://www.diako-leipzig.de, http://www.fz-borstel.de, http://www.klinikum-vest.de, http://www.kerckhoff-klinik.de, http://www.klinikum-offenbach.de, http://kemperhof.gk.de, http://www.klinikum-straubing.de, http://www.klinikum-ingolstadt.de, http://www.klinikum-kulmbach.de, http://www.joho-dortmund.de, http://www.friedrich-ebert-krankenhaus.de/, http://www.prosper-hospital.de, http://www.ortenau-klinikum.de/, http://www.gp-ruesselsheim.de, http://www.kv-keoa.de, https://www.sankt-elisabeth-hospital.de, http://www.evkmh.de, http://www.gzbiwo.de, http://www.sana-hanse-klinikum-wismar.de, http://www.vinzenz-hospital.de, http://www.klinikum-duisburg.de/home.html, http://www.klinik-donaustauf.de, http://www.alb-fils-kliniken.de/, http://www.gesundheitnord.de/krankenhaeuserundzentren/kbm.html, http://www.kk-bochum.de, http://www.klinikum-fulda.de/, http://www.u-e-k.de, http://klinikum-lueneburg.de, http://www.Ketteler-krankenhaus.de, http://www.klinikum-landshut.de, http://www.johanniter-rheinhausen.de, http://www.lukasneuss.de, http://www.meinediakonie.de/evk/index.html, http://www.diako-harz.de/startseite/, http://www.klinikum-ludwigsburg.de, http://www.vinzenz-verbund.de/vinzenz-braunschweig/, http://www.drk-kh-neuwied.de, http://www.donau-isar-klinikum.de, http://www.paracelsus-kliniken.de/osnabrueck, http://www.klinikum-bad-hersfeld.de, http://www.klinikum-ab-alz.de, http://www.kkrn.de, http://www.uniklinik-duesseldorf.de, http://www.hufeland.de, http://www.bethanien-krankenhaus.de/, http://www.Kliniken-Heidenheim.de, http://www.sfh-muenster.de, http://www.diekreiskliniken.de, http://www.krankenhaus-erkelenz.de, http://www.shk-ndh.de, http://www.dritter-orden.de, http://www.leopoldina-krankenhaus.com, http://www.marien-hospital-witten.de, http://www.rotkreuzkliniken.de, http://www.klinikumfrankfurt.de, http://www.evkk.de, http://www.kkh-gummersbach.de, http://www.evkwesel.de, http://www.klinikum-guetersloh.de, http://www.klinikum-goerlitz.de/, http://www.klinikum-ibbenbueren.de, http://www.khdf.de, http://www.barbaraklinik.de, http://www.kk-bottrop.de, http://www.bundeswehrkrankenhaus-ulm.de, http://www.hancken.de/, http://www.sana-hm.de, http://www.stgeorgklinikum.de, http://www.bethesda-wuppertal.de, http://www.klinik-bad-trissl.de, http://www.imland.de, http://www.krupp-krankenhaus.de, http://www.kliniksued-rostock.de, http://www.markus-krankenhaus.de, http://www.kliniken-muehldorf.de, http://glkn.de, http://www.vinzenz-hanau.de, http://www.barmherzige-muenchen.de, http://www.johanna-etienne-krankenhaus.de, http://www.siloah.de, http://www.marien-hospital-dueren.de, http://www.ludmillenstift.de, http://www.k-plus.de, http://www.kliniken-nordoberpfalz.de, http://www.klinloe.de,

http://klinikum-arnsberg.de/, http://www.klinikum-delmenhorst.de, http://www.klinikumsolingen.de, http://www.oberschwabenklinik.de, http://www.sah-eschweiler.de, http://www.BioMed-Klinik.de, http://www.evk-holzminden.de, http://www.theresienkrankenhaus.de, http://www.filderklinik.de, http://www.regiomed-kliniken.de/, http://www.sana-klinikum-remscheid.de/home.html, http://www.mz-ac.de, http://veramed.de, http://www.evk-duesseldorf.de, http://www.klinikum-coburg.de, http://www.klinikum.wolfsburg.de, http://www.kk-ob.de, http://www.nardiniklinikum.de, http://www.sana-benrath.de/home.html, http://www.ekm-gi.de, http://www.ckbm.de, http://www.tumorbiologie-freiburg.de, http://www.sana-luebeck.de, http://www.josef-hospital.de, http://www.stauferklinikum.de, http://www.josefs-hospital.de, http://www.joho.de, http://www.thueringen-kliniken.de, http://www.harzklinikum.com, http://www.kliniken-suedostbayern.de/de/leistungsspektrum/klinikum-traunstein.htm, http://www.khwe.de, http://www.katharinen-hospital.de/, http://www.medizinisches-zentrum.de/, http://www.klksig.de, http://www.franziskus.de, http://www.katholische-kliniken-lahn.de, http://www.klinikum-burgenlandkreis.de, http://www.klinikum-werra-meissner.de, http://www.heidekreis-klinikum.de, http://www.kkh-erfurt.de/, http://www.st-marien-hospital.contilia.de/, http://www.walburga-krankenhaus.de, http://www.juliusspital.de, http://www.kliniken-gz-kru.de, http://www.klinikum-magdeburg.de, http://www.alexianer-berlin-hedwigkliniken.de, http://johanniter.de/einrichtungen/krankenhaus/genthin-stendal/, http://www.k-ob.de, http://www.kliniken-mtk.de, http://www.vph-bensberg.de, http://www.sfh-ahlen.de, http://www.klinikum-brandenburg.de, http://www.alexianer.de, http://www.marien-hospital.com, http://www.klinikum-kf.de/, http://www.evk-witten.de, http://www.bethlehem.de, http://www.paracelsus-krankenhaus.de, http://www.ammerland-klinik.de/, http://www.krankenhaus-neuwittelsbach.de/, http://www.kliniken-leipziger-land.de, http://www.st-antonius-gronau.de, http://www.khdw.de, http://www.klinikum-saarbruecken.de, http://anregiomed.de/ansbach/, http://www.slk-kliniken.de/, http://www.alexianer-krefeld.de, http://www.sana-kl.de, http://www.lvim-pfalz.de, http://kreisklinik-roth.de/, http://www.kk-om.de, http://www.krankenhaus-buchholz.de, http://www.rkn-kliniken.de/, http://www.main-klinik.de, http://www.elbekliniken.de, http://www.kmg-kliniken.de/index.php/akutversorgung/kliniken-der-kmg-kliniken-plc/kmg-klinikum-guestrow/profil-kmg-klinikum-guestrow, http://www.ckq-gmbh.de, http://www.kk-es.de, http://krankenhaus-brake.de, http://www.christophorus-kliniken.de, http://www.krankenhaus-reinbek.de, http://www.krankenhaus-frankenberg.de, http://www.klinikum-saalekreis.de, http://www.ekonline.de, http://www.diakonissen.de/, http://www.st-bernhard-hospital.de/, https://www.waldkrankenhaus.de/, http://www.klinikum-oberberg.de, http://www.oberlausitz-kliniken.de/, http://www.ostalb-klinikum.de, http://st-agnes-bocholt.de, http://www.gemeinschaftskrankenhaus.de, http://www.st-marienkrankenhaus.de, http://www.vincenz.de, http://www.klinikum-memmingen.de, http://www.regiokliniken.de, https://www.sjk.de/, http://www.klinikum-altenburgerland.de, http://www.elblandkliniken.de, http://www.evk-werden.de, http://www.eko.de, http://kreiskliniken-reutlingen.de, https://www.khagatharied.de, http://www.josef.de, http://www.lakumed.de, http://www.marienhaus-klinikum.de, http://www.krankenhaus-heinsberg.de, http://www.klinik-bergedorf.de, http://malteser-krankenhaus-stcarolus.de, http://www.marien-kh-gmbh.de, http://www.altmark-klinikum.de/seiteninfos/startseite/, http://www.havelland-kliniken.de/, http://hetzelstift.de, http://www.kliniken-bc.de/leistungsspektrum/sana-klinikum-biberach.html, http://www.klinikum-weimar.de, http://www.krankenhaus-juelich.de/, http://www.vinzenzkrankenhaus.de, http://www.agaplesion-elisabethenstift.de/, http://www.klinikum-ds.de/home.html, http://www.klinikum-worms.de/index.html, http://www.stiftshospital-andernach.de, http://www.marienkrankenhaus-kassel.de, http://www.dasdiak-klinikum.de, http://www.ev-krankenhaus.de/unser-haus/ueber-uns.html, http://www.sekeutin.de, http://www.diako-augsburg.de, http://www.bundeswehrkrankenhaus-berlin.de, http://www.klinikum-peine.de, http://www.euregio-klinik.de, http://www.rkh-kassel.de/, http://www.klinikum-ffb.de, http://www.st-elisabeth-hospital.de, http://www.klinikum.uni-heidelberg.de, http://malteser-stjohannesstift.de, http://www.mkh-wnd.de, http://www.bergmannstrost.de/, http://www.annahospital.de, http://www.diakonie-klinikum.de, http://www.klinikum-lichtenfels.de, http://klinikum-oldenburg.de/, http://www.klinikum-whv.de, http://www.hunsrueckklinik.de, http://www.rotkreuzklinik-lindenberg.de/gkl_home.html, http://www.krankenhaus-oberndorf.de, http://www.klinikum-niederberg.de, http://www.kreisklinik-ebersberg.de, http://www.krankenhaus-brilon.de, http://www.drk-kliniken-saar.de, http://www.rottalinnkliniken.de/, http://www.verbund-krankenhaus.de/, http://www.kh-luckenwalde.de, http://www.kmh-stadtlohn.de, http://www.klinikum-badsalzungen.de, http://www.stfranziskus.de/, http://www.drk-kh-kirchen.de, http://krankenhaus-linz.de, http://www.caritas-krankenhaus-lebach.de/, http://www.klinikum-erding.de, http://www.kh-pirmasens.de, http://www.krankenhaus-thuine.de, http://www.sana-gerresheim.de, http://www.sk-mg.de, http://www.vkkd-kliniken.de, http://www.elisabeth-klinikum.de/, http://www.frg-kliniken.de, http://www.klinikum-friedrichshafen.de, http://www.krankenhaus-bietigheim.de, http://www.dreifaltigkeits-hospital.de, http://www.kreisklinik-woerth.de/, http://www.hohenlind.de, http://www.gz-w.de/standorte-betriebsstaetten/buergerhospital-friedberg.html, http://www.krankenhaus-korbach.de/, http://www.bethesda-mg.de, http://www.kkh-hagen.de, http://www.klinikum-emden.de, http://www.klinik-wartenberg.de, http://www.krankenhaus-johanneum.de, http://www.josephstift-dresden.de, http://www.kreiskrankenhaus-wolgast.de, http://www.rkk-apolda.de, http://www.krankenhaus-hermeskeil.de, http://www.ilm-kreis-kliniken.de, http://www.marienhaus-klinikum-saar.de, http://www.kh-muldental.de, http://rechbergklinik.de, http://www.diakonissenhaus.de, http://www.klinikum-nf.de, http://www.kreisklinik-bad-neustadt.de, http://www.st-vinzenz-hospital.de, http://www.krankenhaus-geilenkirchen.de, http://www.ek-leipzig.de, http://ruedersdorf.immanuel.de, http://www.antonius-koeln.de, http://www.zollernalb-klinikum.de, http://www.krankenhaus-puettlingen.de, http://www.segebergerkliniken.de, http://www.klinikum-msp.de, http://www.kkh-demmin.de, http://www.mh-ml.de, http://oevk-trier.de, http://www.hospitalgesellschaft.de, http://www.tropenklinik.de, http://www.d-k-h.de, http://www.donkliniken.de, http://www.hospital-zum-heiligen-geist.de/, http://klinikum-leer.de, http://www.kkh-rotenburg.de, http://www.kkh-stl.de, http://www.drk-chemnitz.de/kontakt/, http://www.erzgebirgsklinikum.de/, http://klinikum-obergoeltzsch.de, http://www.khporz.de, http://www.hospital-greiz.de, http://www.marienhaus-klinikum-eifel.de, http://www.bodden-kliniken.de, http://www.awogsd.de, http://www.klinikum-niederlausitz.de, http://www.marienhospital.de/de/home, http://www.bgu-ludwigshafen.de/, http://www.bonifatius-hospital-lingen.de, http://www.diakonie-klinikum.com, http://www.krankenhaus-kirchlinde.de, http://www.klinikum-altmuehlfranken.de, http://www.kreiskliniken-darmstadt-dieburg.de/, http://www.krankenhaus-wesseling.de, http://www.klinikum-tut.de, http://www.pfeiffersche-stiftungen.de, http://klinikumstadtsoest.de, http://www.marienstift-braunschweig.de, https://www.rems-murr-kliniken.de, http://www.st-katharinen-hospital.de/, http://www.ukb.de, http://hospital-leer.de, http://www.evangelischeskrankenhaus.de, http://www.krankenhaus-remagen.de, http://www.marienhospital-oelde.de, http://www.roteskreuzkrankenhaus.de, http://www.ct-west.de/st-augustinus-krankenhaus-dueren/, http://www.drk-kh-mv.de, http://www.fuerst-stirum-klinik.de, http://www.hgk-koeln.de/, http://www.johanniter-krankenhaus.de, http://www.barmherzige-schwandorf.de/, http://www.kkhviechtach.de, http://www.kkh-freiberg.com/, http://www.lmkgmbh.de, http://www.skh-ft.de, http://www.klinikum-landsberg.de, http://www.evk-mettmann.de, http://www.kreiskrankenhaus-alsfeld.de, http://www.gross-sand.de, http://www.dominikus-berlin.de, http://www.heilig-geist-hospital.de, http://www.kksaar.de, http://www.drk-kh-alzey.de.drktg.de/az, http://www.klinikumhalle.de, http://www.krankenhaus-wermelskirchen.de, http://www.mhb-bottrop.de, http://www.josef-krankenhaus.de, http://www.bathildis.de, http://www.hjk-muenster.de/index.htm, http://www.klinikum-starnberg.de, http://klinik-vincentinum.de, https://www.kh-neuwerk.de, http://www.sana-ruegen.de, http://www.stiftungsklinik-weissenhorn.de, http://www.rkk-klinikum.de, http://www.wmk-hvb.de, http://www.evk-haspe.de, http://www.klinikum-ld-suew.de, http://www.kreisklinik-gg.de, http://www.os-kh.de, http://www.krankenhaus-pruem.de, http://www.warnowklinik-buetzow.de, http://www.franziskushospital.de, http://www.kkh-bergstrasse.de, http://www.krankenhaus-emmendingen.de, http://www.mkh-bgl.de/, http://www.hospital-borken.de, http://www.sankt-marien-ratingen.de, http://www.drk-kh-altenkirchen.de, http://www.drk-krankenhaus.de, http://www.gk-bonn.de, http://www.klinikum-fichtelgebirge.de, http://www.sjs-bremen.de/, http://www.kreiskliniken-unterallgaeu.de, http://www.kreiskrankenhaus-osterholz.de, http://www.marien-kh.de, http://www.drk-kh-hachenburg.de, http://www.klinikum-doebeln.de, http://www.klf-web.de, http://krankenhaus-rotthalmuenster.de, http://klinikum-dresden.de, http://www.diakonissenkrankenhaus-dresden.de, http://elbe-elster-klinikum.de, http://www.evk-castrop-rauxel.de, http://www.ik-h.de/home.html, http://www.kkimk.de, http://www.augustinum-kliniken.de, http://www.mkkliniken.de, http://www.marienhaus-klinikum-ahr.de, http://www.vincentius-hd.de, http://www.naemi-wilke-stift.de, http://ostemed.de, http://www.pleissental-klinik.de, http://www.raphaelsklinik.de, http://krankenhaus-nettetal.de, http://www.st-irmgardis.de/de/startseite.html, http://www.krankenhaus-varel.de, http://www.st-josefs.de, http://www.stmartinus-langenfeld.de, http://www.theresien-krankenhaus.de, http://www.krankenhaus-tutzing.de, http://www.kreuznacherdiakonie.de, http://www.evk.de, http://www.gz-odw.de/index.php, http://www.katharinen-hospiz.de, http://www.krankenhaus-spremberg.de, http://www.hospital-schleiz.de, http://www.kreiskrankenhaus-weisswasser.de/, http://www.sana-huerth.de, http://krankenhaus-beckum.de, http://www.einbecker-buergerspital.de, http://www.kh-nuernberger-land.de, http://www.kkhzwiesel.de/, http://www.kreiskrankenhaus-hoechstadt.de, http://www.marienhospital-bruehl.de, http://www.mathilden-hospital.de, http://www.alexianer-diepholz.de, http://www.eichsfeld-klinikum.de, http://www.st-brigida.de, http://www.diakoniekrankenhaus.de,

http://www.heh-bs.de info@heh-bs.de, http://www.erlabrunn.de, http://www.klinik-preetz.de/, http://www.krankenhaus-damme.de, http://www.kreisklinik-wolfratshausen.de, http://www.saarpfalz-kreis.de/gesundheitspark/index.html, http://caritas-klinik-pankow.de, http://www.krankenhaus-bethel.de, http://www.hdz-nrw.de, http://www.juedisches-krankenhaus.de, http://www.diekliniken.de, http://www.krankenhaus-mainburg.de, http://www.khtbb.de/khtbb/index.php, http://www.oberhavel-kliniken.de, http://www.Klinik-Schindlbeck.de, http://www.johannes-krankenhaus.com, http://www.alexianer-potsdam.de/home/, http://www.buergerhospital-ffm.de, http://www.diakoniekrankenhaus-henriettenstiftung.de, http://www.ekh-luckau.de, http://www.evangelisches-johannesstift.de, http://www.marien-hospital-bonn.de, http://www.muellerklinik.de/, http://www.jhwaf.de, http://www.k-kl.de, http://www.donauklinik-neu-ulm.de, http://www.martin-luther-krankenhaus-bo.de, http://www.drk-biedenkopf.de, http://www.kkh-mek.de, http://www.lausitzklinik.de, http://www.malteser-krankenhaus-berlin.de, https://sana-klinik-nuernberg.de, http://www.sana-radevormwald.de, http://www.sankt-josef-hospital.de/, http://www.wertachkliniken.de, http://www.grn.de, http://www.krankenhaus-doberan.de, http://www.bethelnet.de, http://www.marienhausklinik-ottweiler.de, http://www.sana-pegnitz.de, http://www.klinik-ellwangen.de, http://www.clemens-hospital.de/, http://www.krankenhaus-rodalben.de, http://www.sjs-del.de, http://www.adk-gmbh.de, http://www.caritasstjosef.de/, http://www.DiakonieNeuendettelsau.de, http://www.diakomed.de, http://www.dkh-wehrda.de/startseite, http://www.bfh.drk-tb.de/, http://www.keh-berlin.de/de/index, http://www.goldbergklinik.de, http://www.hassberg-kliniken.de, http://www.kirchbergklinik.de, http://www.kkh-stadthagen.de, http://www.malteser-krankenhaus-stjohannes.de, http://malteser-stjosef.de, http://www.marienerft.de, http://www.marienhospital-letmathe.de, http://www.rotkreuzkliniken-sued.de, http://www.diako-kassel.de, http://www.dkhseehausen.de, http://www.lutherstiftung.de, http://www.krankenhaus-bonn.de, http://www.k-k-o.de, http://www.kliniken-st-elisabeth.de/, http://www.klinik-fraenkische-schweiz.de, http://www.klipa.de, http://www.kh-wtm.de, http://www.krankenhaus-weilburg.de, http://www.illertalklinik-illertissen.de, http://marienhospital-darmstadt.de, http://www.marien-krankenhaus-brandenburg.de, http://www.rochus-hospital.de/, http://www.bethanien-heidelberg.de, http://www.bethesda-ulm.de, http://www.eduardus.de, http://www.hospital-fritzlar.de, http://www.krankenhaus-prignitz.de, http://www.kliniken-hochfranken.de, http://www.josephinum.de, http://www.augusta-duesseldorf.de, http://www.krankenhaus-plettenberg.de, http://kh-gmbh-ws.de, http://www.krankenhausgruenstadt.de, http://www.klinikum-crailsheim.de, http://www.kliniken-schwesternschaft-muenchen.de, http://www.khehst.de, http://www.st.josef-kh.de, http://www.stmarienkrankenhaus.de, http://www.agaplesion-diakoniekrankenhaus-ingelheim.de, http://www.alexianer-toenisvorst.de/home/, http://www.aller-weser-klinik.de, http://www.egzb.de, http://www.krankenhaus-dierdorf-selters.de, http://www.elsey.de, http://www.ilmtalklinik.de, http://www.karl-olga-krankenhaus.de/home.html, http://www.kkh-rinteln.de, http://www.eichhof-online.de, http://krankenhaus-vilshofen.de, http://www.krankenhaus-waltershausen-friedrichroda.de, http://www.krankenhaus-linnich.de/, http://www.smh-luedinghausen.de/de/startseite.html, http://www.krankenhaus-eisenberg.de, http://www.alice-hospital.de, http://www.ameos.de, http://el-stift.de/krankenhaus/, http://www.drk-kh-gvm.de, http://www.geomed-klinik.de, http://www.herz-jesu-krankenhaus.de, http://www.kliniken-nea.de, http://www.klinik-tt.de, http://www.klinikum-forchheim.de, http://www.klinik-st-blasien.de, http://www.mariannen-hospital.de, http://www.khsc.de, http://jokba.de, http://www.marienkrankenhaus-berlin.de, http://www.vinzenz-klinik.de/, http://www.cura.org, http://www.dkd-dessau.de, http://www.kamillus-klinik.de, https://www.clementinenhaus.de, http://www.krankenhaus-woltersdorf.de, http://www.berglandklinik.de, http://www.krankenhaus-sachsenhausen.de, http://www.marien-hospital-papenburg.de/, http://www.marienkrankenhaus-cochem.de, http://www.eks-schwerte.de, http://www.salzachklinik-fridolfing.de, http://www.sankt-katharinen-ffm.de/, http://www.anna-klinik.de, http://www.medbo.de, http://www.diako-krankenhaus.de, http://www.dominikus.de, http://www.fabricius-klinik.de, http://www.krankenhaus-geseke.de, http://www.klinik-bogen.de, http://schlosspark-klinik.de/, http://www.kh-as.de, http://st-elisabeth-sz.de, http://www.st-hubertusstift.de, http://sankt-josef-werden.de, http://www.evk-koeln.de, http://www.franziskus-berlin.de/, http://www.gesundheitszentrum-glantal.de, http://www.hgh-bensheim.de, http://www.hk-gmbh.net, http://www.huettenhospital.de, http://www.krankenhaus-naturheilweisen.de, http://www.krankenhaus-neuenbuerg.de, http://www.loreley-kliniken.de, http://www.luisen-krankenhaus.de/, http://www.marienhauskliniken.de, http://www.sana-wildbad.de, http://www.sankt-gertrauden.de, http://www.vincentius-speyer.de, http://www.sankt-vinzenz.de, http://www.scivias-caritas.de, http://www.kh-lengenfeld.de, http://www.gesundheitszentrum-winterberg.de, http://www.bethanien-chemnitz.de, http://www.awo-khbg.de, http://www.diak-ka.de, http://www.drk-klinik-kaufungen.de, http://www.krankenhaus-grimmen.de, http://www.elisabeth-krankenhaus-ge.de/, http://www.krankenhaus-muehlacker.de, http://www.altersmedizin-potsdam.de, http://www.klinik-eichstaett.de/, http://www.klinik-fuessen.de/, http://www.klinik-mallersdorf.de, http://www.klinikum-penzberg.de, http://www.krankenhaus-marbach.de, http://www.krankenhaus-warstein.de, http://www.neumariahilf.de, http://krankenhaus-rheiderland.de/index.php?id=2, http://www.guterhirte-ludwigshafen.de, http://www.neckar-odenwald-kliniken.de, http://www.piushospital.com, http://www.saarlandkliniken.de, http://www.stadtklinik-werdohl.de, http://ihr-gesundheitszentrum.de, http://www.krankenhaus-eitorf.de, http://marienstift-friesoythe.de, http://www.drk-kh-diez.de, http://www.gz-treuchtlingen.de, http://klinikum-osnabrueckerland.de, http://www.schreiberklinik.de, http://www.klinikevb.de, http://www.klinik-buchloe.de/, http://klinikum-gap.de, http://www.14-nothelfer.de, http://www.krankenhaus-stockach.de, http://www.waldfriede.de, http://www.kreiskrankenhaus-wasserlos.de/, http://www.kkh-sob.de, http://www.hans-prinzhorn-klinik.de/, http://www.klinikum-karlsbad.de, http://www.kurpfalzkrankenhaus.de, http://www.elisabeth-dortmund.de, http://www.st.josefskrankenhaus.de, http://www.bk-marsberg.de, http://www.uhz.de, https://www.vitanas.de/de/klinische_centren/geratie_maerkisches_viertel/krankenhaus_geriatrie_maerkisches_viertel.php, http://www.diakonissen-krankenhaus.de/, http://bgu-murnau.de, http://www.diakoniewerk-muenchen.de, http://www.diana-klinik.de, http://www.drk-middelburg.de, http://www.bethanien-iserlohn.de, http://www.sozialwerk-meiningen.de, http://www.hgz-bb.de, http://www.herzzentrum-dresden.com, http://geriatrie-ratzeburg.de/, http://www.krankenhaus-neustadt.de, http://www.christophsbad.de, http://www.juraklinik-schesslitz.de, http://www.krankenhaus-norderney.de/, http://krankenhaus-salem.de, http://www.krankenhaus-st-josef-wuppertal.de/, http://www.krankenhaus-vaihingen.de, http://www.kkh-parsberg.de, http://www.maria-theresia-klinik.de, http://www.marien-hospital-dortmund.de, http://www.krankenhaus-am-crivitzer-see.de, http://www.otto-fricke-krankenhaus.de, http://www.rku.de, http://www.skh-arnsdorf.sachsen.de, http://www.sana-kt.de, http://sb.shg-kliniken.de/index.php?id=1810, http://www.krankenhaus-lohne.de, http://www.kh-acura-kliniken.com, http://www.kinderkrankenhaus.net, http://www.bak-schneeberg.de, https://www.klinik-dr-fruehauf.de, http://www.evang-krankenhaus-regensburg.de/, http://www.ekh-gesundbrunnen.de, http://www.ortho-klinik.de, http://www.fachklinik-enzensberg.de, http://www.fachklinik-dietenbronn.de, http://www.fachkliniken-radeburg.de, http://www.fkh-hubertusburg.de, http://www.geriatrische-fachklinik-rheinhessen-nahe.de, http://www.hkz-rotenburg.de, http://www.isarklinikum.de, http://www.kav-krankenhaus.de, http://www.erler-klinik.de/, http://www.klinik-hallerwiese.de, http://www.khv-ha-wa.de, http://www.krankenhaus-werneck.de, http://www.krankenhaus-rummelsberg.de, http://www.krankenhaus-st-camillus.de, http://www.st-marienstift.de, http://krankenhaus-wegscheid.de/, http://www.klinikum-duesseldorf.lvr.de, http://www.lwl-klinik-lengerich.de, http://www.median-kliniken.de, http://www.nierenzentrum-heidelberg.com, http://www.privatklinik-hellge.com, http://www.rhein-mosel-fachklinik-andernach.de/, http://www.st-anna-klinik.de, http://www.st-rochus-dieburg.de, http://www.vinzenz-altena.de, http://www.VvPH.de, http://www.vulpiusklinik.de

E

SONSTIGES

E.1 E-MAIL ALEXA

E-Mail Antwort Alexa Internet (support@alexa.zendesk.com) vom 10. Juli 2015:

> Hi Tim,
>
> Unfortunately we do not offer historical top sites lists. However, if you have specific data you'd like to get data for we do have data going back to 2007 available via our APIs: http://aws.amazon.com/alexa.
> Cheers,
> Debbie
> Alexa Customer Support

E.2 VERSIONSSTÄNDE DER GENUTZTEN PYTHON-BIBLIOTHEKEN

- alembic==0.8.8
- androguard==3.0
- asn1crypto==0.22.0
- beautifulsoup4==4.5.3
- Brotli==0.6.0
- certifi==2017.4.17
- cffi==1.10.0
- chardet==2.3.0
- click==6.6
- colorama==0.3.7
- cryptography==1.9
- Cuckoo==2.0.3
- Django==1.8.4
- django-extensions==1.6.7
- dpkt==1.8.7
- ecdsa==0.13
- elasticsearch==5.3.0
- enum34==1.1.6
- Flask==0.10.1
- Flask-SQLAlchemy==2.1
- functools32==3.2.3.post2
- HTTPReplay==0.2.1
- idna==2.5
- ipaddress==1.0.18
- itsdangerous==0.24
- Jinja2==2.8
- jsbeautifier==1.6.2
- jsonschema==2.6.0
- M2Crypto==0.24.0
- Mako==1.0.6
- MarkupSafe==1.0
- olefile==0.43
- oletools==0.42
- peepdf==0.3.6
- pefile2==1.2.11
- Pillow==3.2.0
- pkg-resources==0.0.0
- pycparser==2.17
- pycrypto==2.6.1
- pymisp==2.4.54
- pymongo==3.0.3
- pyOpenSSL==17.0.0
- python-dateutil==2.4.2
- python-editor==1.0.3
- python-magic==0.4.12
- pythonaes==1.0
- pyvmomi==6.5.0.2017.5.post1
- requests==2.13.0
- scapy==2.3.2
- SFlock==0.2.16
- six==1.10.0
- SQLAlchemy==1.0.8
- tlslite==0.4.9
- tlslite-ng==0.8.0a1
- urllib3==1.21.1
- wakeonlan==0.2.2
- Werkzeug==0.12.2
- yara-python==3.5.0

[1] G. Acar, M. Juárez, N. Nikiforakis, C. Díaz, S. F. Gürses, F. Piessens und B. Preneel. FPDetective: dusting the web for fingerprinters. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 1129–1140, 2013. doi: 10.1145/2508859.2516674. URL http://doi.acm.org/10.1145/2508859.2516674.

[2] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan und C. Diaz. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 674–689, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2957-6.

[3] L. Agarwal, N. Shrivastava, S. Jaiswal und S. Panjwani. Do Not Embarrass: Re-examining User Concerns for Online Tracking and Advertising. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 8:1–8:13, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2319-2. doi: 10.1145/2501604.2501612.

[4] R. Albert und A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan. 2002. doi: 10.1103/RevModPhys.74.47. URL http://link.aps.org/doi/10.1103/RevModPhys.74.47.

[5] Alphabet Inc. Werbeumsätze von Google in den Jahren 2001 bis 2016 (in Milliarden US-Dollar), 2016. URL https://de.statista.com/statistik/daten/studie/75188/umfrage/werbeumsatz-von-google-seit-2001/.

[6] A. Alsaid. Detecting Web Bugs with Bugnosis: Privacy Advocacy through Education. In D. M. M. Jr., editor, *Privacy Enhancing Technologies, Second International Workshop, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers*, pages 13–26, 2002. doi: 10.1007/3-540-36467-6_2.

[7] A. Alsaid und D. Martin. Detecting Web Bugs with Bugnosis: Privacy Advocacy Through Education. In *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 13–26, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-00565-X.

[8] B. Antoniewicz. Windows DLL Injection Basics, 2013. URL http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html. Letzter Zugriff am 24.02.2018.

[9] M. Ayenson, D. J. Wambach, A. Soltani, N. Good und C. J. Hoofnagle. Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning. *Social Science Research Network Working Paper Series*, July 2011.

[10] M. Backes, A. Kate, M. Maffei und K. Pecina. ObliviAd: Provably Secure and Practical Online Behavioral Advertising. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 257–271, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4681-0. doi: 10.1109/SP.2012.25.

[11] R. Balebako, P. G. Leon, R. Shay, B. Ur, Y. Wang und L. F. Cranor. Measuring the effectiveness of privacy tools for limiting behavioral advertising. In *In Web 2.0 Workshop on Security and Privacy*, 2012.

[12] A. Barth. HTTP State Management Mechanism. RFC 6265 (Proposed Standard), Apr. 2011. URL http://www.ietf.org/rfc/rfc6265.txt. Letzter Zugriff am 25.02.2018.

[13] A. Barth. The Web Origin Concept. RFC 6454, Dec. 2011. URL http://www.ietf.org/rfc/rfc6454.txt. Letzter Zugriff am 25.02.2018.

[14] J. Bau, J. Mayer, H. Paskov und J. C. Mitchell. A promising direction for web tracking countermeasures. In *Proceedings of Web 2.0 Security and Privacy (W2SP)*, 2013.

[15] M. Belshe, R. Peon und M. Thomson. Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540, Nov. 2015. URL http://www.ietf.org/rfc/rfc7540.txt.

[16] E. Beltermann. Benutzerverfolgung durch staatliche Websites, Juli 2015. URL https://netzpolitik.org/2015/benutzerverfolgung-durch-staatliche-websites/. Letzter Zugriff am 24.02.2018.

[17] C. J. Bennett. Cookies, web bugs, webcams and cue cats: Patterns of surveillance on the world wide web. *Ethics and Information Technology*, 3(3):195–208, 2001. ISSN 1572-8439. doi: 10.1023/A:1012235815384. URL http://dx.doi.org/10.1023/A:1012235815384.

[18] T. Berners-Lee, L. Masinter und M. Mccahill. RFC 1738: Uniform resource locator (URL), 1994. URL https://www.ietf.org/rfc/rfc1738.txt.

[19] T. Berners-Lee, R. Fielding und H. Nielsen. RFC 1945 – Hypertext Transfer Protocol – HTTP/1.0, May 1996. URL http://www.ietf.org/rfc/rfc1945.txt.

[20] T. Berners-Lee, R. Fielding und L. Masinter. Rfc 3986, uniform resource identifier (uri): Generic syntax. Request For Comments (RFC), 2005. URL http://www.ietf.org/rfc/rfc3986.txt.

[21] M. Bilenko, M. Richardson und J. Tsai. Targeted, not tracked: Client-side solutions for privacy-friendly behavioral advertising, 2011.

[22] G. Blokdijk. *Virtualization - The Complete Cornerstone Guide to Virtualization Best Practices: Concepts, Terms, and Techniques for Successfully Planning, Implementing and Managing Enterprise IT Virtualization Technology - Second Edition.* Emereo Publishing, 2012. ISBN 9781486434534.

[23] T. Bläsing, L. Batyuk, A. D. Schmidt, S. A. Camtepe und S. Albayrak. An android application sandbox system for suspicious software detection. In *2010 5th International Conference on Malicious and Unwanted Software*, pages 55–62, Oct 2010. doi: 10.1109/MALWARE.2010.5665792.

[24] K. Boda, A. M. Földes, G. G. Gulyás und S. Imre. User Tracking on the Web via Cross-browser Fingerprinting. In *Proceedings of the 16th Nordic Conference on Information Security Technology for Applications*, NordSec'11, pages 31–46, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-29614-7. doi: 10.1007/978-3-642-29615-4_4.

[25] B. Bollobás. *Random Graphs.* Cambridge University Press, Cambridge, 2 edition, 008 2001. ISBN 9780511814068. doi: 10.1017/CBO9780511814068. URL https://www.cambridge.org/core/books/random-graphs/E21023008001CFA182CE666F5028489F.

[26] U. Brandes und T. Erlebach. *Network analysis. Methodological foundations.* Springer, Berlin; New York, 2005.

[27] J. Bremer. x86 API Hooking Demystified, 2015. URL http://jbremer.org/x86-api-hooking-demystified. Letzter Zugriff am 24.02.2018.

[28] T. Bujlow, V. Carela-Español, J. Solé-Pareta und P. Barlet-Ros. Web Tracking: Mechanisms, Implications, and Defenses. *CoRR*, abs/1507.07872, 2015.

[29] Bundesamt für Sicherheit in der Informationstechnik. BSI-Grundschutz Katalog, 2017. URL http://www.bsi.de/gshb/deutsch/index.htm. Letzter Zugriff am 01.02.2017.

[30] A. Chaabane, M. A. Kaafar und R. Boreli. Big Friend is Watching You: Analyzing Online Social Networks Tracking Capabilities. In *Proceedings of the 2012 ACM Workshop on Workshop on Online Social Networks*, WOSN '12, pages 7–12, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1480-0. doi: 10.1145/2342549.2342552.

[31] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[32] E. Y. Chen, J. Bau, C. Reis, A. Barth und C. Jackson. App Isolation: Get the Security of Multiple Browsers with Just One. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*,

CCS '11, pages 227–238, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0948-6. doi: 10.1145/2046707.2046734. URL http://doi.acm.org/10.1145/2046707.2046734.

[33] M. Christodorescu, S. Jha, D. Maughan, D. Song und C. Wang. *Malware Detection*. Advances in Information Security. Springer US, 2007. ISBN 9780387445991.

[34] M. Costa, D. Gomes und M. J. Silva. The evolution of web archiving. *International Journal on Digital Libraries*, pages 1–15, 2016. ISSN 1432-1300. doi: 10.1007/s00799-016-0171-9. URL http://dx.doi.org/10.1007/s00799-016-0171-9.

[35] P. Crowder und D. Crowder. *Creating Web Sites Bible*. Bible. Wiley, 2008. ISBN 9780470372593.

[36] Cuckoo Foundation. Cuckoo Sandbox, 2018. URL https://cuckoosandbox.org/. Letzter Zugriff am 24.02.2018.

[37] Cuckoo Foundation. Cuckoo Sandbox Book, 2018. URL http://docs.cuckoosandbox.org/en/2.0.3/. Letzter Zugriff am 24.02.2018.

[38] N. Daswani, A. Ranadive, S. Rizvi, M. Gagnon, T. Demir und G. Eisenhaur. Behavioral scanning of mobile applications, Aug. 12 2014. US Patent 8,806,647.

[39] A. Datta, M. C. Tschantz und A. Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *CoRR*, abs/1408.6491, 2014. URL http://arxiv.org/abs/1408.6491.

[40] A. Datta, M. C. Tschantz und A. Datta. Automated experiments on ad privacy settings. *PoPETs*, 2015(1):92–112, 2015.

[41] M. Day. Preserving the fabric of our lives: A survey of web preservation initiatives. In *International Conference on Theory and Practice of Digital Libraries*, pages 461–472. Springer, 2003.

[42] H. V. de Sompel, M. Nelson und R. Sanderson. HTTP Framework for Time-Based Access to Resource States – Memento. RFC 7089, Dec. 2013. URL http://www.ietf.org/rfc/rfc7089.txt.

[43] M. Duerst und M. Suignard. RFC 3987: Internationalized Resource Identifiers (IRIs). RFC 3987 (Proposed Standard), see http://www.ietf.org/rfc/rfc3987.txt, 1 2005. URL http://www.ietf.org/rfc/rfc3987.txt.

[44] P. Eckersley. How Unique is Your Web Browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[45] C. Eckert. *IT-Sicherheit - Konzepte, Verfahren, Protokolle (6. Aufl.)*. Oldenbourg, 2009. ISBN 978-3-486-58999-3. URL http://www.oldenbourg-wissenschaftsverlag.de/olb/de/1.c.1679883.de.

[46] M. Egele, T. Scholte, E. Kirda und C. Kruegel. A survey on automated dynamic malware-analysis techniques and tools. *ACM Comput. Surv.*, 44(2):6:1–6:42, Mar. 2008. ISSN 0360-0300. doi: 10.1145/2089125.2089126.

[47] K. Egevang und P. Francis. RFC 1631 The IP Network Address Translator (NAT), May 1994. URL http://tools.ietf.org/html/rfc1631.

[48] S. Englehardt und A. Narayanan. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1388–1401, 2016. doi: 10.1145/2976749.2978313.

[49] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. Mayer, A. Narayanan und E. W. Felten. Cookies That Give You Away: The Surveillance Implications of Web Tracking. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 289–299, 2015. doi: 10.1145/2736277.2741679.

[50] D. Esposito. *Modern Web Development: Understanding domains, technologies, and user experience*. Developer Reference. Pearson Education, 2016. ISBN 9781509300549.

[51] C. Evans, C. Palmer und R. Sleevi. Public Key Pinning Extension for HTTP. RFC 7469, Apr. 2015. URL http://www.ietf.org/rfc/rfc7469.txt. Letzter Zugriff am 25.02.2018.

[52] M. Falahrastegar, H. Haddadi, S. Uhlig und R. Mortier. *The Rise of Panopticons: Examining Region-Specific Third-Party Web Tracking*, pages 104–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-54999-1. doi: 10.1007/978-3-642-54999-1_9.

[53] E. W. Felten und M. A. Schneider. Timing attacks on web privacy. In *Proceedings of the 7th ACM Conference on Computer and Communications Security*, CCS '00, pages 25–32, New York, NY, USA, 2000. ACM. ISBN 1-58113-203-4. doi: 10.1145/352600.352606. URL http://doi.acm.org/10.1145/352600.352606.

[54] O. Ferrand. How to detect the cuckoo sandbox and to strengthen it? *J. Computer Virology and Hacking Techniques*, 11(1):51–58, 2015. doi: 10.1007/s11416-014-0224-9.

[55] R. Fielding, J. Gettys, J. Mogul, H. Frystyk und T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068 (Proposed Standard), 1 1997. URL http://www.ietf.org/rfc/rfc2068.txt. Obsoleted by RFC 2616.

[56] R. T. Fielding und J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. RFC 7230, June 2014. URL http://www.ietf.org/rfc/rfc7230.txt. Letzter Zugriff am 28.02.2018.

[57] R. T. Fielding und J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests. RFC 7232, June 2014. URL http://www.ietf.org/rfc/rfc7232.txt. Letzter Zugriff am 19.02.2018.

[58] D. Fifield und S. Egelman. *Fingerprinting Web Users Through Font Metrics*, pages 107–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-662-47854-7. doi: 10.1007/978-3-662-47854-7_7. URL http://dx.doi.org/10.1007/978-3-662-47854-7_7.

[59] M. Fredrikson und B. Livshits. Repriv: Re-imagining content personalization and in-browser privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 131–146, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4402-1. doi: 10.1109/SP.2011.37.

[60] N. Fruchter, H. Miao, S. Stevenson und R. Balebako. Variations in Tracking in Relation to Geographic Location. *CoRR*, abs/1506.04103, 2015. URL http://arxiv.org/abs/1506.04103.

[61] H. Gersdorf, B. Paal und R. Bornemann. *Informations- und Medienrecht: GRC, EMRK, GG, RStV, BGB, IFG, VIG, GWB, TKG, TMG : Kommentar*. C.H. Beck, 2014. ISBN 9783406661969. URL https://beck-online.beck.de/?typ=reference&y=100&g=TMG&p=5.

[62] Ghostery, Inc. Ghostery - home page. https://www.ghostery.com/, 2015. Letzter Zugriff am 24.02.2018.

[63] I. Goldberg, D. Wagner, R. Thomas, E. A. Brewer et al. A secure environment for untrusted helper applications: Confining the wily hacker. In *Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, volume 6, pages 1–1, 1996.

[64] C. Greamo und A. Ghosh. Sandboxing and virtualization: Modern tools for combating malware. *IEEE Security Privacy*, 9(2):79–82, March 2011. ISSN 1540-7993. doi: 10.1109/MSP.2011.36.

[65] L. G. Greenwald und T. J. Thomas. Toward undetected operating system fingerprinting. In *Proceedings of the First USENIX Workshop on Offensive Technologies*, WOOT '07, pages 6:1–6:10, Berkeley, CA, USA, 2007. USENIX Association.

[66] A. Grosskurth und M. W. Godfrey. A Reference Architecture for Web Browsers. In *21st IEEE International Conference on Software Maintenance (ICSM 2005), 25-30 September 2005, Budapest, Hungary*, pages 661–664, 2005. doi: 10.1109/ICSM.2005.13.

[67] S. Guha, B. Cheng und P. Francis. Privad: Practical privacy in online advertising. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 169–182, Berkeley, CA, USA, 2011. USENIX Association.

[68] D. Gupta und B. Mehte. Forensics analysis of sandboxie artifacts. In *International Symposium on Security in Computing and Communication*, pages 341–352. Springer, 2013.

[69] S. Hackett, B. Parmanto und X. Zeng. A retrospective look at website accessibility over time. *Behaviour & Information Technology*, 24(6): 407–417, 2005. doi: 10.1080/01449290500066661.

[70] A. A. Hagberg, D. A. Schult und P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, Aug. 2008.

[71] C. Hauschke. Third-Party-Elemente in deutschen Bibliothekswebseiten. *Informationspraxis*, 2(2), 2016.

[72] C. Heilmann und M. Francis. *Web Development Solutions: Ajax, APIs, Libraries, and Hosted Services Made Easy*. Apress, 2007. ISBN 9781430203933.

[73] P. Helm. Group privacy in times of big data. a literature review. *Digital Culture & Society*, 2(2):137–152, 2016.

[74] A. R. Hevner, S. T. March, J. Park und S. Ram. Design Science in Information Systems Research. *MIS Q.*, 28(1):75–105, Mar. 2004. ISSN 0276-7783.

[75] A. Hidayat. PhantomJS | PhantomJS, 2016. URL http://phantomjs.org/. Letzter Zugriff am 24.02.2018.

[76] R. M. Hinden und D. S. E. Deering. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Dec. 1998. URL http://www.ietf.org/rfc/rfc2460.txt.

[77] H. Hockx-Yu. The Past Issue of the Web. In *Proceedings of the 3rd International Web Science Conference*, WebSci '11, pages 12:1–12:8, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0855-7. doi: 10.1145/2527031.2527050.

[78] J. Hodges, C. Jackson und A. Barth. HTTP Strict Transport Security (HSTS). RFC 6797, Nov. 2012. URL http://www.ietf.org/rfc/rfc6797.txt.

[79] G. Hoglund und J. Butler. *Rootkits: Subverting the Windows Kernel*. Addison-Wesley Software Security Series. Addison-Wesley, 2006. ISBN 9780321294319.

[80] A. Hosseini. Ten Process Injection Techniques: A Technical Survey of Common and Trending Process Injection Techniques, 2017. URL https://www.endgame.com/blog/technical-blog/ten-process-injection-techniques-technical-survey-common-and-trending-process. Letzter Zugriff am 24.02.2018.

[81] IIPC. International Internet Preservation Consortium, 2016. URL http://netpreserve.org/. Letzter Zugriff am 24.02.2018.

[82] G. Inc. Chromium - the chromium projects, 2018. URL https://www.chromium.org/Home. Letzter Zugriff am 24.02.2018.

[83] D. INOUE, K. YOSHIOKA, M. ETO, Y. HOSHIZAWA und K. NAKAO. Automated malware analysis system and its sandbox for revealing malware's internal and external activities. *IEICE Transactions on Information and Systems*, E92.D(5):945–954, 2009. doi: 10.1587/transinf.E92.D.945.

[84] Internet Archive. Frequently Asked Questions, 2017. URL https://archive.org/about/faqs.php. Letzter Zugriff am 24.02.2018.

[85] Internet Engineering Task Force. Internet Protocol. RFC 791, Sept. 1981. URL http://www.ietf.org/rfc/rfc791.txt.

[86] Internet Live Stats. Anzahl der Internetnutzer weltweit in den Jahren 1997 bis 2015 sowie eine Prognose für 2016 (in Millionen), 2014. URL https://de.statista.com/statistik/daten/studie/186370/umfrage/anzahl-der-internetnutzer-weltweit-zeitreihe/.

[87] C. Jackson, A. Bortz, D. Boneh und J. C. Mitchell. Protecting browser state from web privacy attacks. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 737–744, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135884. URL http://doi.acm.org/10.1145/1135777.1135884.

[88] A. Janc und L. Olejnik. Web Browser History Detection As a Real-world Privacy Threat. In *Proceedings of the 15th European Conference on Research in Computer Security*, ESORICS'10, pages 215–231, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15496-4, 978-3-642-15496-6.

[89] D. Jang, R. Jhala, S. Lerner und H. Shacham. An empirical study of privacy-violating information flows in JavaScript web applications. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 270–283, 2010. doi: 10.1145/1866307.1866339.

[90] B. Jansen. *Handbook of Research on Web Log Analysis*. IGI Global, 2008. ISBN 9781599049755.

[91] R. Jansen, F. Tschorsch, A. Johnson und B. Scheuermann. The sniper attack: Anonymously deanonymizing and disabling the tor network. Technical report, OFFICE OF NAVAL RESEARCH ARLINGTON VA, 2014.

[92] B. Kahle. HTTP Archive - Trends, 2016. URL http://httparchive.org/trends.php. Letzter Zugriff am 24.02.2018.

[93] S. Kamkar. evercookie - virtually irrevocable persistent cookies, 2010. URL http://samy.pl/evercookie/. Letzter Zugriff am 24.02.2018.

[94] M. Kranch und J. Bonneau. Upgrading https in mid-air: An empirical study of strict transport security and key pinning, 2015. URL ftp://ftp.cs.princeton.edu/techreports/2015/986.pdf. Letzter Zugriff am 24.02.2018.

[95] M. Krieger. On the Value of Online User Behaviour – A Supporting Software Framework, 2017.

[96] B. Krishnamurthy und C. E. Wills. Privacy diffusion on the web: A longitudinal perspective. In *In Procs World Wide Web Conference*, page 09, 2009.

[97] B. Krishnamurthy, K. Naryshkin und C. Wills. Privacy leakage vs. protection measures: the growing disconnect. In *Proceedings of the Web*, volume 2, pages 1–10, 2011.

[98] S. Krishnan und F. Monrose. Dns prefetching and its privacy implications: When good things go bad. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

[99] D. M. Kristol und L. Montulli. HTTP State Management Mechanism. RFC 2109, 1998. URL https://tools.ietf.org/html/rfc2109.txt.

[100] R. Lämmel und E. Pek. Understanding privacy policies - A study in empirical analysis of language usage. *Empirical Software Engineering*, 18(2):310–374, 2013. doi: 10.1007/s10664-012-9204-1.

[101] P. Leon, B. Ur, R. Shay, Y. Wang, R. Balebako und L. Cranor. Why Johnny Can'T Opt out: A Usability Evaluation of Tools to Limit Online Behavioral Advertising. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 589–598, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207759.

[102] P. G. Leon, L. F. Cranor, A. M. McDonald und R. McGuire. Token attempt: The misrepresentation of website privacy policies through the misuse of p3p compact policy tokens. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '10,

pages 93–104, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0096-4. doi: 10.1145/1866919.1866932.

[103] A. Lerner, A. K. Simpson, T. Kohno und F. Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12*, 2016. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/lerner.

[104] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, Feb. 1966.

[105] T. Li, H. Hang, M. Faloutsos und P. Efstathopoulos. TrackAdvisor: Taking Back Browsing Privacy from Third-Party Trackers. In *PAM*, volume 8995 of *Lecture Notes in Computer Science*, pages 277–289. Springer, 2015.

[106] T. Libert. Privacy implications of health information seeking on the web. *Commun. ACM*, 58(3):68–77, 2015. doi: 10.1145/2658983.

[107] T. Libert. Exposing the Hidden Web: An Analysis of Third-Party HTTP Requests on 1 Million Websites. *CoRR*, abs/1511.00619, 2015. URL http://arxiv.org/abs/1511.00619.

[108] Library of Congress. Library of Congress Collections Policy Statements Supplementary Guidelines, 2016. URL https://www.loc.gov/acq/devpol/webarchive.pdf. Letzter Zugriff am 24.02.2018.

[109] Lightbeam. Addon for Firefox. https://addons.mozilla.org/en-US/firefox/addon/lightbeam/, 2015. Letzter Zugriff am 24.02.2018.

[110] lucb1e. Lucb1e.com :: Cookieless cookies, 2013. URL http://lucb1e.com/rp/cookielesscookies/. Letzter Zugriff am 24.02.2018.

[111] M. Maaß, P. Wichmann, H. Pridöhl und D. Herrmann. Privacyscore: Improving privacy and security via crowd-sourced benchmarks of websites. In *Privacy Technologies and Policy - 5th Annual Privacy Forum, APF 2017, Vienna, Austria, June 7-8, 2017, Revised Selected Papers*, pages 178–191, 2017. doi: 10.1007/978-3-319-67280-9_10.

[112] D. Malandrino, A. Petta, V. Scarano, L. Serra, R. Spinelli und B. Krishnamurthy. Privacy Awareness About Information Leakage: Who Knows What About Me? In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, pages 279–284, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2485-4.

[113] J. L. Marill, A. Boyko, M. Ashenfelder und L. Graham. Tools and techniques for harvesting the World Wide Web. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 403–403. ACM, 2004.

[114] D. Martin, H. Wu und A. Alsaid. Hidden Surveillance by Web Sites: Web Bugs in Contemporary Use. *Commun. ACM*, 46(12):258–264, Dec. 2003. ISSN 0001-0782. doi: 10.1145/953460.953509.

[115] J. R. Mayer und J. C. Mitchell. Third-party web tracking: Policy and technology. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 413–427, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4681-0. doi: 10.1109/SP.2012.47.

[116] D. McCoy, K. Bauer, D. Grunwald, T. Kohno und D. Sicker. Shining light in dark places: Understanding the tor network. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 63–76. Springer, 2008.

[117] A. M. Mcdonald, L. F. Cranor, A. M. Mcdonald und L. F. Cranor. A Survey of the Use of Adobe Flash Local Shared Objects to Respawn HTTP Cookies, 2011.

[118] J. P. S. Medeiros, A. M. Brito und P. S. Motta Pires. *An Effective TCP/IP Fingerprinting Technique Based on Strange Attractors Classification*, pages 208–221. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-11207-2. doi: 10.1007/978-3-642-11207-2_16.

[119] J. Mikians, L. Gyarmati, V. Erramilli und N. Laoutaris. Detecting price and search discrimination on the internet. In *11th ACM Workshop on Hot Topics in Networks, HotNets-XI, Redmond, WA, USA - October 29 - 30, 2012*, pages 79–84, 2012. doi: 10.1145/2390231.2390245.

[120] S. Miller. *Piwik Web Analytics Essentials*. Community experience distilled. Packt Publishing, 2012. ISBN 9781849518499.

[121] J. Mogul, L. M. Masinter, R. T. Fielding, J. Gettys, P. J. Leach und T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, Mar. 2013. URL http://www.ietf.org/rfc/rfc2616.txt.

[122] G. Mohr, M. Stack, I. Rnitovic, D. Avery und M. Kimpton. Introduction to heritrix. In *4th International Web Archiving Workshop*, 2004.

[123] D. Mortensen. *Yahoo! Web Analytics: Tracking, Reporting, and Analyzing for Data-Driven Insights*. Serious Skills. Wiley, 2009. ISBN 9780470524091.

[124] K. Mowery und H. Shacham. Pixel perfect: Fingerprinting canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.

[125] K. Mowery, D. Bogenreif, S. Yilek und H. Shacham. Fingerprinting information in JavaScript implementations. In H. Wang, editor, *Proceedings of W2SP 2011*. IEEE Computer Society, May 2011.

[126] M. Mulazzani, M. Huber, M. Leithner und S. Schrittwieser. Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting, 2013.

[127] D. T. Narten, R. P. Draves und S. Krishnan. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941, Sept. 2007. URL http://www.ietf.org/rfc/rfc4941.txt.

[128] D. T. Narten, T. Jinmei und D. S. Thomson. IPv6 Stateless Address Autoconfiguration. RFC 4862, Sept. 2007. URL http://www.ietf.org/rfc/rfc4862.txt.

[129] Network Working Group. RFC 2617, HTTP Authentication: Basic and Digest Access Authentication. *http://www.ietf.org/rfc/rfc2617.txt*, 1999.

[130] M. Neugschwandtner, P. M. Comparetti und C. Platzer. Detecting malware's failover c&c strategies with squeeze. In *Proceedings of the 27th annual computer security applications conference*, pages 21–30. ACM, 2011.

[131] Nielsen. Bruttowerbeaufwendungen in den Printmedien in Deutschland in den Jahren 2013 bis 2015 (in Milliarden Euro), 2016. URL https://de.statista.com/statistik/daten/studie/154896/umfrage/werbeaufwendungen-in-printmedien-seit-2005/.

[132] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. V. Acker, W. Joosen, C. Kruegel, F. Piessens und G. Vigna. You are what you include: large-scale evaluation of remote javascript inclusions. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 736–747, 2012. doi: 10.1145/2382196.2382274.

[133] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens und G. Vigna. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 541–555, 2013. doi: 10.1109/SP.2013.43.

[134] R. Nithyanand, S. Khattak, M. Javed, N. Vallina-Rodriguez, M. Falahrastegar, J. E. Powles, E. D. Cristofaro, H. Haddadi und S. J. Murdoch. Ad-blocking and counter blocking: A slice of the arms race. *CoRR*, abs/1605.05077, 2016. URL http://arxiv.org/abs/1605.05077.

[135] M. Nottingham. Web Linking. RFC 5988, Oct. 2010. URL http://www.ietf.org/rfc/rfc5988.txt.

[136] G. C. Obasuyi und A. Sari. Security challenges of virtualization hypervisors in virtualized hardware environment. *International Journal of Communications, Network and System Sciences*, 8(07):260, 2015.

[137] L. Olejnik, C. Castelluccia und A. Janc. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, Vigo, Spain, July 2012. URL https://hal.inria.fr/hal-00747841.

[138] L. Olejnik, G. Acar, C. Castelluccia und C. Díaz. The Leaking Battery - A Privacy Analysis of the HTML5 Battery Status API. In *DPM/QASA@ESORICS*, volume 9481 of *Lecture Notes in Computer Science*, pages 254–263. Springer, 2015. ISBN 978-3-319-29882-5. URL http://dblp.uni-trier.de/db/conf/esorics/dpm2015.html#OlejnikACD15.

[139] X. Pan, Y. Cao und Y. Chen. I Do Not Know What You Visited Last Summer: Protecting users from stateful third-party web tracking with TrackingFree browser. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, 2015.

[140] A. Panchenko, L. Niessen, A. Zinnen und T. Engel. Website Fingerprinting in Onion Routing Based Anonymization Networks. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, pages 103–114, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1002-4. doi: 10.1145/2046556.2046570.

[141] R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, J. M. Pedersen und A. Czech. Analysis of malware behavior: Type classification using machine learning. In *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–7, June 2015. doi: 10.1109/CyberSA.2015.7166115.

[142] G. J. Popek und R. P. Goldberg. Formal requirements for virtualizable third generation architectures. *Commun. ACM*, 17(7):412–421, July 1974. ISSN 0001-0782. doi: 10.1145/361011.361073.

[143] M. Portnoy. *Virtualization Essentials*. Wiley, 2016. ISBN 9781119267720.

[144] J. Postel. User Datagram Protocol. RFC 768, Aug. 1980. URL http://www.ietf.org/rfc/rfc768.txt.

[145] J. Postel. Transmission Control Protocol. RFC 793, Sept. 1981. URL http://www.ietf.org/rfc/rfc793.txt.

[146] PricewaterhouseCoopers. Umsätze mit Onlinewerbung in Deutschland in den Jahren 2005 bis 2021* (in Millionen Euro), 2016. URL https://de.statista.com/statistik/daten/studie/165473/umfrage/umsatzentwicklung-von-onlinewerbung-seit-2005/.

[147] A. Provataki und V. Katos. Differential malware forensics. *Digit. Investig.*, 10(4):311–322, Dec. 2013. ISSN 1742-2876.

[148] N. Provos, D. McNamee, P. Mavrommatis, K. Wang und N. Modadugu. The ghost in the browser analysis of web-based malware. In *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, pages 4–4, Berkeley, CA, USA, 2007. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1323128.1323132.

[149] B. Rector und J. Newcomer. *Win32 Programming*. Number Bd. 1 in Addison-Wesley advanced Windows series. Addison-Wesley Developers Press, 1997. ISBN 9780201634921.

[150] J. Reschke. Initial Hypertext Transfer Protocol (HTTP) Authentication Scheme Registrations. RFC 7236, June 2014. URL http://www.ietf.org/rfc/rfc7236.txt.

[151] E. Rescorla und T. Dierks. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, Aug. 2008. URL http://www.ietf.org/rfc/rfc5246.txt. Letzter Zugriff am 03.02.2018.

[152] A. Reznichenko, S. Guha und P. Francis. Auctions in do-not-track compliant internet advertising. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 667–676, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0948-6. doi: 10.1145/2046707.2046782.

[153] F. Roesner, T. Kohno und D. Wetherall. Detecting and Defending Against Third-party Tracking on the Web. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 12–12, Berkeley, CA, USA, 2012. USENIX Association.

[154] S. Rose, M. Larson, D. Massey, R. Austein und R. Arends. Resource Records for the DNS Security Extensions. RFC 4034, Mar. 2005. URL http://www.ietf.org/rfc/rfc4034.txt. Letzter Zugriff am 25.02.2018.

[155] M. E. Russinovich, D. A. Solomon und A. Ionescu. *Windows Internals, Part 1: Covering Windows Server 2008 R2 and Windows 7*. Microsoft Press, 6th edition, 2012. ISBN 0735648735, 9780735648739.

[156] M. E. Russinovich, D. A. Solomon und A. Ionescu. *Windows Internals, Part 2: Covering Windows Server 2008 R2 and Windows 7 (Windows Internals)*. Microsoft Press, 2012. ISBN 0735665877, 9780735665873.

[157] P. Saint-Andre und D. J. C. Klensin. Uniform Resource Names (URNs). RFC 8141, Apr. 2017. URL http://www.ietf.org/rfc/rfc8141.txt.

[158] K. A. Scarfone, M. P. Souppaya und P. Hoffman. Sp 800-125. guide to security for full virtualization technologies. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States, 2011.

[159] S. Schelter und J. Kunegis. Tracking the Trackers: A Large-Scale Analysis of Embedded Web Trackers. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[160] M. Schmiedecker, P. Reschl, M. Huber, M. Leithner, S. Schrittwieser und E. Weippl. Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting. In *Web 2.0 Workshop on Security and Privacy (W2SP)*, 5 2013.

[161] M. Schneider, M. Enzmann und M. Stopczynski. Web-Tracking-Report 2014. Technical Report SIT-TR-2014-01, Fraunhofer-Institut für Sichere Informationstechnologie, Feb. 2014.

[162] SeleniumHQ. Selenium - Web Browser Automation, 2016. URL http://docs.seleniumhq.org/. Letzter Zugriff am 24.02.2018.

[163] M. K. Shankarapani, S. Ramamoorthy, R. S. Movva und S. Mukkamala. Malware detection using assembly and API call sequences. *Journal in Computer Virology*, 7(2):107–119, 2011. doi: 10.1007/s11416-010-0141-5.

[164] K. Sigurðsson. Incremental crawling with heretrix. In *Proceedings of the 5th International Web Archiving Workshop (IWAW'05)*, Vienna, Austria, 2005. URL http://iwaw.europarchive.org/05/papers/iwaw05-sigurdsson.pdf.

[165] M. Sikorski und A. Honig. *Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software*. No Starch Press, 2012. ISBN 9781593274306.

[166] H. A. Simon. *The Sciences of the Artificial (3rd Ed.)*. MIT Press, Cambridge, MA, USA, 1996. ISBN 0-262-69191-4.

[167] R. Simon. *Windows NT Win32 API SuperBible*. Other Sams Series. Waite Group Press, 1997. ISBN 9781571690890.

[168] A. Soltani, S. Canty, Q. Mayo, L. Thomas und C. J. Hoofnagle. Flash Cookies and Privacy. In *Intelligent Information Privacy Management, Papers from the 2010 AAAI Spring Symposium, Technical Report SS-10-05, Stanford, California, USA, March 22-24, 2010*, 2010. URL http://www.aaai.org/ocs/index.php/SSS/SSS10/paper/view/1070.

[169] A. K. Sood und R. J. Enbody. Malvertising–exploiting web advertising. *Computer Fraud & Security*, 2011(4):11–16, 2011.

[170] K. Soska und N. Christin. Automatically detecting vulnerable websites before they turn malicious. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 625–640, San Diego, CA, Aug. 2014. USENIX Association. ISBN 978-1-931971-15-7. URL https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/soska.

[171] M. Spaniol und G. Weikum. Tracking Entities in Web Archives: The LAWA Project, 2012.

[172] M. Spreitzenbarth, F. Freiling, F. Echtler, T. Schreck und J. Hoffmann. Mobile-sandbox: Having a deeper look into android applications. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 1808–1815, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1656-9. doi: 10.1145/2480362.2480701.

[173] M. Spreitzenbarth, T. Schreck, F. Echtler, D. Arp und J. Hoffmann. Mobile-sandbox: combining static and dynamic analysis with machine-learning techniques. *International Journal of Information Security*, 14(2):141–153, 2015.

[174] J. M. Stewart, E. Tittel und M. Chapple. *CISSP: Certified Information Systems Security Professional Study Guide*. SYBEX Inc., Alameda, CA, USA, 4th edition, 2008. ISBN 0470276886, 9780470276884.

[175] M. Stopczynski und M. Zugelder. Reducing User Tracking through Automatic Web Site State Isolations. In *Information Security - 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings*, pages 309–327, 2014. doi: 10.1007/978-3-319-13257-0_18.

[176] M. Stopczynski und M. Zugelder. *Reducing User Tracking through Automatic Web Site State Isolations*, pages 309–327. Springer International Publishing, Cham, 2014. ISBN 978-3-319-13257-0. doi: 10.1007/978-3-319-13257-0_18.

[177] L. Story. To aim ads, web is keeping closer eye on you, 3 2008. URL http://www.nytimes.com/2008/03/10/technology/10privacy.html. Letzter Zugriff am 25.02.2018.

[178] N. Taffin. LAWA – longitudinal analytics of web archive data, 2016. URL http://www.lawa-project.eu/.

[179] A. Tanenbaum. *Moderne Betriebssysteme*. Pearson Studium - IT. Pearson Deutschland, 2009. ISBN 9783827373427.

[180] A. Tatnall. *Web Technologies: Concepts, Methodologies, Tools, and Applications*. Contemporary research in information science and technology. Information Science Reference, 2010. ISBN 9781605669823.

[181] A. Team. The early days of web analytics, 2015. URL https://amplitude.com/blog/2015/06/15/the-early-days-of-web-analytics/. Letzter Zugriff am 24.02.2018.

[182] The Hollywood Reporter. Prognose der Online-Werbeumsätze weltweit in den Jahren 2014 bis 2018 (in Milliarden US-Dollar), 2016. URL https://de.statista.com/statistik/daten/studie/237596/umfrage/prognose-zur-entwicklung-der-online-werbeumsaetze-weltweit/.

[183] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum und S. Barocas. Adnostic: Privacy preserving targeted advertising. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2010, San Diego, California, USA, 28th February - 3rd March 2010*, 2010. URL http://www.isoc.org/isoc/conferences/ndss/10/pdf/05.pdf.

[184] M. Toyoda und M. Kitsuregawa. The History of Web Archiving. *Proceedings of the IEEE*, 100:144–1443, 13 2012. ISSN 0018-9219. doi: 10.1109/JPROC.2012.2189920.

[185] TrackingObserver. TrackingObserver: A Browser-Based Web Tracking Detection Platform, 2015. URL http://trackingobserver.cs.washington.edu/. Letzter Zugriff am 24.02.2018.

[186] H. T. T. Truong, E. Lagerspetz, P. Nurmi, A. J. Oliner, S. Tarkoma, N. Asokan und S. Bhattacharya. The company you keep: Mobile malware infection rates and inexpensive risk indicators. In *Proceedings of the 23rd international conference on World wide web*, pages 39–50. ACM, 2014.

[187] V. Vaishnavi und W. Kuechler. Design Research in Information Systems. Letztes Update am 11. November 2012., Jan. 2004. URL http://www.desrist.org/design-research-in-information-systems/.

[188] M. Vasilescu, L. Gheorghe und N. Tapus. Practical malware analysis based on sandboxing. In *2014 RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference*, pages 1–6, Sept 2014. doi: 10.1109/RoEduNet-RENAM.2014.6955304.

[189] T. Wambach. Dynamische Trackererkennung im Web durch Sandbox-Verfahren. In P. Schartner, K. Lemke-Rust und M. Ullmann, editors, *D.A.CH Security 2015*, pages 301–310, 2015.

[190] T. Wambach. Ökonomisierung von Nutzerverhalten – historische Entwicklung und aktueller Stand. *Forschungsjournal Soziale Bewegungen*, 30(2):162–169, 2017.

[191] T. Wambach und K. Bräunlich. Retrospective Study of Third-party Web Tracking. In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016), Rome, Italy, February 19-21, 2016.*, pages 138–145, 2016. doi: 10.5220/0005741301380145.

[192] T. Wambach und K. Bräunlich. The Evolution of Third-Party Web Tracking. In *International Conference on Information Systems Security and Privacy*, pages 130–147. Springer, 2016.

[193] T. Wambach und K. Knorr. Technische Prüfung der Datenschutzerklärungen auf deutschen Hochschulwebseiten. *FHWS Science Journal*, 3 (2015)(2):44 – 57, 2016.

[194] T. Wambach, L. Schulte und K. Knorr. Einbettung von Drittinhalten im Web. *Datenschutz und Datensicherheit*, 40(8):523–527, 2016. doi: 10.1007/s11623-016-0650-6. URL http://dx.doi.org/10.1007/s11623-016-0650-6.

[195] S. D. Warren und L. D. Brandeis. The right to privacy. *Harward Law Review*, 4(5):193–220, December 1890.

[196] G. Weikum, N. Ntarmos, M. Spaniol, P. Triantafillou, A. A. Benczúr, S. Kirkpatrick, P. Rigaux und M. Williamson. Longitudinal Analytics on Web Archive Data: It's About Time! In *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings*, pages 199–202, 2011. URL http://www.cidrdb.org/cidr2011/Papers/CIDR11_Paper26.pdf.

[197] Z. Weinberg, E. Y. Chen, P. R. Jayaraman und C. Jackson. I Still Know What You Visited Last Summer: Leaking Browsing History via User Interaction and Side Channel Attacks. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 147–161, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4402-1. doi: 10.1109/SP.2011.23.

[198] A. Westin. *Privacy and Freedom.* Bodley Head, 1970. ISBN 9780370013251.

[199] C. Willems, T. Holz und F. Freiling. Toward automated dynamic malware analysis using cwsandbox. *IEEE Security Privacy*, 5(2):32–39, March 2007. ISSN 1540-7993. doi: 10.1109/MSP.2007.45.

[200] C. E. Wills und C. Tatar. Understanding What They Do with What They Know. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, WPES '12, pages 13–18, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1663-7. doi: 10.1145/2381966.2381969.

[201] B. Witt. *Datenschutz an Hochschulen: ein Praxishandbuch für Deutschland am Beispiel der Universitäten Baden-Württembergs.* LegArtis, 2004. ISBN 9783936494365.

[202] R. Wojtczuk und R. Kashyap. The sandbox roulette: Are you ready for the gamble? *Black Hat Europe 2013*, 414:800–125, 2013.

[203] Q. Wu, Q. Liu, Y. Zhang, P. Liu und G. Wen. *A Machine Learning Approach for Detecting Third-Party Trackers on the Web*, pages 238–258. Springer International Publishing, Cham, 2016. ISBN 978-3-319-45744-4. doi: 10.1007/978-3-319-45744-4_12.

[204] X. Xing, W. Meng, B. Lee, U. Weinsberg, A. Sheth, R. Perdisci und W. Lee. Understanding malvertising through ad-injecting browser extensions. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1286–1295, Republic and Canton

of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-3469-3. doi: 10.1145/2736277.2741630.

[205] T. Yen, Y. Xie, F. Yu, R. P. Yu und M. Abadi. Host Fingerprinting and Tracking on the Web: Privacy and Security Implications. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*, 2012.

CURRICULUM VITÆ

PERSÖNLICHE DATEN

	Tim Wambach * 25.10.1985 in Trier
Anschrift	Kronprinzenstr. 2, 40217 Düsseldorf
Familienstand	ledig

BERUFLICHE ERFAHRUNG

seit 2018	Referent bei der Landesbeauftragten für Datenschutz und Informationsfreiheit NRW
2014 - 2018	Wissenschaftlicher Mitarbeiter an der Universität Koblenz–Landau
2004 - 2018	Softwareentwickler der Gameforge AG in Karlsruhe
2014/2015	Lehrauftrag für „IT-Sicherheit“ im Bachelorstudiengang an der Hochschule Trier
2012 - 2014	Associate Security Consultant der Siemens AG in München
2010 - 2012	Assistent im Fachbereich Informatik an der Fachhochschule Trier
2011	Wissenschaftlicher Mitarbeiter an der Fachhochschule Trier
2008/2009	Studentische Hilfskraft an der Fachhochschule Trier
2004/2005	Praktikum am Finanzamt Trier

SCHULBILDUNG UND STUDIUM

2009 - 2011	Studium der Informatik (M.Sc.) an der Fachhochschule Trier
2005 - 2009	Studium der Informatik (B.Sc.) an der Fachhochschule Trier
2005	Fachhochschulreife
2002 - 2004	Höhere Berufsfachschule an der BBS Gewerbe und Technik in Trier
1996 - 2002	Robert-Schuman-Realschule in Trier
1992 - 1996	Matthias-Grundschule in Trier

1. November 2018